UNIVERSITY OF OSLO
Department of informatics

Design of environmental and mismatch self-compensating, low jitter delay lines in 90nm low power CMOS

Master thesis

Morten Hamremoen Røed

14.08.2012

Abstract

This thesis explores the use of digitally controllable delay elements to make up delay lines that can compensate for the increased problems with environmental, mismatch and process-variations. This becomes more and more important as the transistor sizes continuously are being scaled as the technology evolves and short channel effects begin to dominate. In radar systems the variations become even more significant as the propagated pulse travels at the speed of light and the resolution is confined by these variations. Even variations as small as $10ps$ will limit the resolution in the time domain to $3mm$.

The main sources of and solutions to reduce the mechanisms behind the variations are discussed. 3 different delay lines with different delay element architectures are compared with respect to tunability, resolution, area, power consumption, linearity, temperature variations and noise (e.g. jitter). A proposed solution to create an even more temperature independent digitally controllable delay element using back gate tuning is also presented.

Acknowledgments

This thesis is submitted to the Department of Informatics, Faculty of Mathematics and Natural Science at University of Oslo as a part of the M.Sc. degree in Nano and-microelectronics.

I would like to express my gratitude to my internal supervisor Tor Sverre Lande for always coming up with suggestions for new ideas to test and the freedom of choice.

I also owe a great thanks to my external supervisors Olav and Kristian at Novelda for the proposed thesis, educational meetings and helpful advice as well as new ideas to work on.

My friends Tor-Eivind, Geir, Christoffer and Michael has contributed to a great work environment and I would like to thank them for many great times studying at the lab, playing billiards and at other social events. It has been a lot of fun and a memorable time of my life.

Last but not least I would also like to show my appreciation to Cathrine for always believing in me and supporting me, making my work a whole lot easier. Thank you.

Table of contents

List of Figures

1 Introduction

Accurate and programmable delay elements are widely used in applications such as $PLL's$, $DLL's$, $DCO's$ and impulse based radar systems. Inserting a shunt capacitor, variable resistor or applying current starving to the delay elements are techniques that will change a buffer's internal rise and/or fall time [1], and thus the propagation delay of the element.

The cascaded delay elements such as binary weighted multiplexed delay lines and tapped delay lines [2][2] are heavily influenced by mismatch, process, temperature and power supply-variations, resulting in a need for reliable techniques to measure and compensate for these variations. If the delay elements on chip are measurable and tunable, the delay elements can be calibrated to have the desired propagation delay, only confined by the resolution of the measurements and the tunability of the elements. Therefore digitally controllable delay elements ($DCDEs$) are required for accurate characterization and optimal performance.

1.1 Problem Statement

- How to design environmental and mismatch self-compensating, low jitter delay lines in $90nm$ low power $CMOS$?

1.2 Approach

To approach the problem statement, more specifically "how to design environmental and mismatch self-compensating, low jitter delay lines in $90nm$ low power $CMOS$", several combinations of two (or more) types of delay elements forming a single delay line will be investigated and designed. The performance of the different types of delay elements will then be compared and evaluated relative to each other and existing designs. If one of the combinations of delay elements results in a high performance delay line, the design might be a candidate for the next architecture to be incorporated in the next version of the Novelda Impulse Radar.

2 Methods

As mentioned previously, the timing calibration of delay elements in Silicon with short delay intervals is hard to accomplish due to the extensive influence of process, voltage and temperature variations, in combination with layout issues. This extensive influence is due to the fact that small transistor sizes are utilized to minimize capacitance, resulting in high resolution (short time interval) delay elements with low power consumption. A tradeoff is that short channel effects begin to dominate and confines the accuracy of the tunable delay elements. This results in a need for reliable techniques to compensate for these variations. This is what brings us to the question; what if two or more types of tunable delay elements were incorporated to make up a delay line? Could this provide us with the best of two (or more) worlds, resulting in higher accuracy delay lines that can compensate for environmental and process variations, or will the digitally controllable delay elements introduce other disadvantages?

Several problems with existing designs and proposed solutions are discussed in more detail in the succeeding chapters.

2.1 Process Variations and Mismatch

Process variations or static variations are a natural variation in transistor device parameters that occurs when integrated circuits are fabricated due to limitations in the manufacturing process, such as process tolerances and mask misalignment. Process variations are becoming increasingly important as the channel lengths of $CMOS$ devices are decreasing, due to the fact that the variation becomes a larger percentage of the full device length or width. Some device features are approaching some fundamental dimensions such as the size of atoms (oxide thickness) and the usable wavelength of light used in the lithography for fabrication. Examples of process variations are:

- Doping concentration
- Oxide thickness
- Diffusion depths
- Transistor lengths
- Transistor widths
- Lateral diffusion (figure 2.1 a)
- Overetching (figure 2.1 b)
- Material imperfections

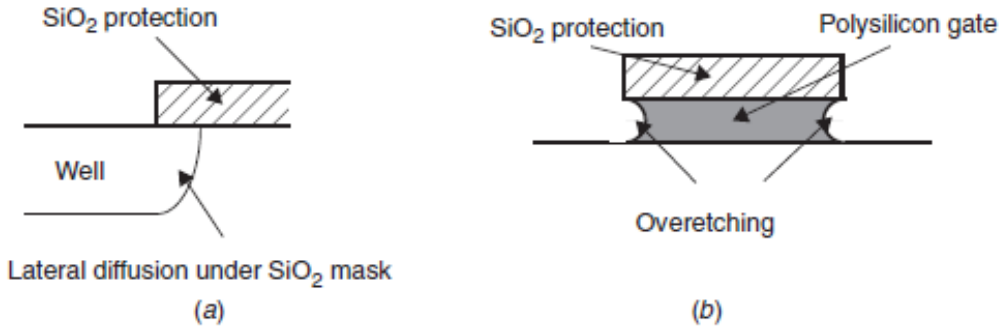

Figure 2.1. Various two-dimensional effects causing mismatch [3]

Changes in these device parameters due to process variations will result in mismatch, which is especially important for analog circuits. Mismatch describes the fact that devices from the same wafer can and will have different electrical parameters; therefore two identically designed devices will never be achieved in the fabrication process. This is referred to as random mismatch and is what is usually thought of as matching. The random mismatch has a Gaussian distribution as seen in figure 2.2 [4].

Another form of mismatch is the systematic mismatch which is caused by the designer, including design errors and poor layout.

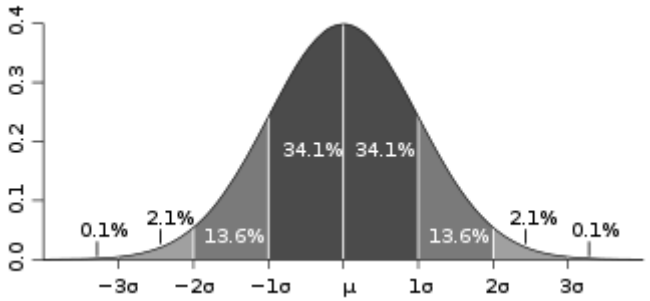

Figure 2.2. Gaussian distribution function for threshold voltage mismatches [5]

Since the resolution of the fabrication process is limited to the wavelength of usable light used in the lithography, or the refraction index or the fluid in immersion lithography [6], it is intuitive that the degree of mismatch increases with the decreasing transistor sizes. The degree of mismatch, here as a function of the threshold voltage, is inversely proportional with the square root of the transistor area as shown in equation 2.1 [7]. Note that to it takes 4 times the silicon area to reduce the mismatch by a factor of $1/2$.

$$\boldsymbol{\sigma}_{\Delta V_{th}} = \frac{A_{V_{th}}}{\sqrt{W \cdot L}} \tag{2.1}$$

where $A_{V_{th}}$ is the mismatch constant equal to the standard deviation between the threshold voltage of a pair of transistors each sized 1 *square micron, W is the transistor width and L is the transistor length.*

Variations in these electrical parameters, such as threshold voltage and sheet resistance, will result in variations in device performance. This may reduce the yield of the batch if the performance does not meet the specifications.

To take these process variations into account while designing, Monte Carlo simulations should be used to simulate over a wide range of randomly chosen device parameters or corners. This is shown in figure 2.3 which clearly indicates where the name “corners” has its origin.

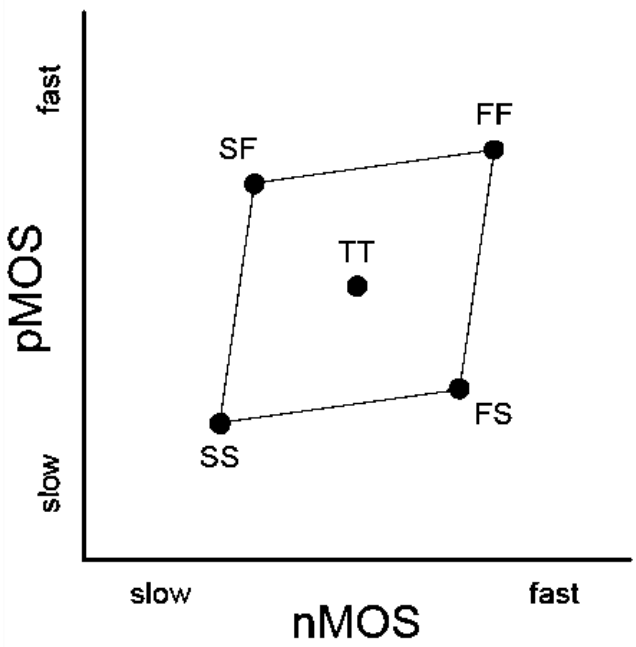

Figure 2.3. Different simulation corners for MOSFET’s [8]

From figure 2.3 the parameters S, T and F refers to Slow, Typical and Fast device parameters respectively. The first letter denotes the $nMOS$ device parameter while the second letter denotes the $pMOS$ device parameter.

Fast transistor parameters include:

- Short L_{eff} and/or Wide W_{eff}
- Low threshold voltage V_t
- Thin t_{OX} (Oxide Thickness)

The parameters L_{eff} and W_{eff} refer to the actual transistor length and width as they differ from the ideally designed values. The effective transistor length and width can be expressed by equation 2.2 and 3 [9]:

$$L_{eff} = L + X_L - 2 \cdot L_D \quad (2.2)$$

$$W_{eff} = W + X_W - 2 \cdot W_D \quad (2.3)$$

where L and W is the ideally designed transistor length and width, X_L and X_W is the factor by which the length and width differs from the ideal, L_D is the source and drain lateral diffusion and W_D is the bulk diffusion.

Slow transistor device parameters are the opposite of fast transistor parameters and typical parameters is somewhere in the middle. Slow transistor parameters are helpful to examine the worst case simulations and evaluate the design margin against the specifications for the circuit, e.g. characterization. It is to be noted that worst case considerations should be the benchmark to achieve reliability for all conditions.

Since the process variations are static, they can be compensated for more easily than environmental variations. This can be done by carefully biasing the devices to calibrate them to a certain delay relative to an external reference.

2.2 Environmental Variations

Delay elements are also heavily influenced by environmental variations such as temperature and power supply variations as well as random jitter.

Complementary Metal-Oxide-Semiconductor ($CMOS$) is known to have inferior noise performance compared to the traditional bipolar or junction gate field-effect transistor ($JFET$) technology. As an example a $CMOS$ operational amplifier ($OPAMP$) has around two to three orders of magnitude worse noise performance than a bipolar or $JFET\ OPAMP$ [10]. This noise will affect the jitter performance of the circuit.

Figure 2.4 shows a noise model for the main noise sources for different circuit elements [11]. Note that capacitors and inductors are not included as they do not generate noise, they only transmit it.

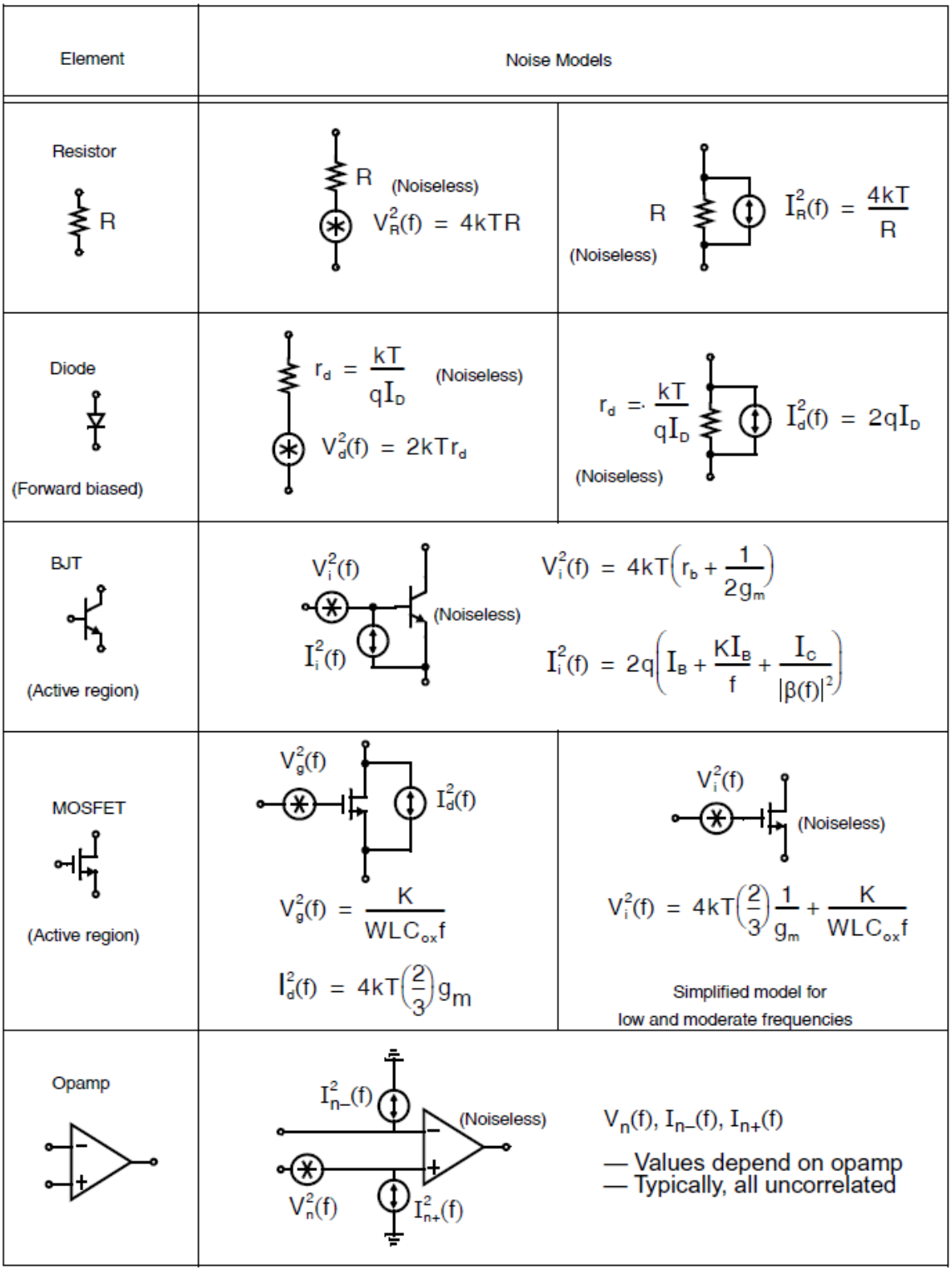

Figure 2.4. Circuit elements and their noise models [11]

2.2.1 Jitter

Jitter is the undesired delay variation of an assumed periodical signal in a series of edges propagating down the delay line as shown in figure 2.5. It can produce bit errors in a digital system that interrupts the logic functions as pulses may miss the sampling point. Jitter limits the resolution of the delay lines and as the clock frequencies increase the timing jitter has increasingly tighter limitations [12] thus making it more important to consider in fast state-of-the-art processes. The jitter performance is affected by several sources:

- Thermal noise (Nyquist noise or Johnson noise)
- Flicker noise
- Shot noise
- Coupling noise
- Process, Voltage and Temperature variations (PVT)
- Layout
- Pulse Repetition Frequency (PRF)
- Electro Magnetic Interference (EMI)
- Slew Rate

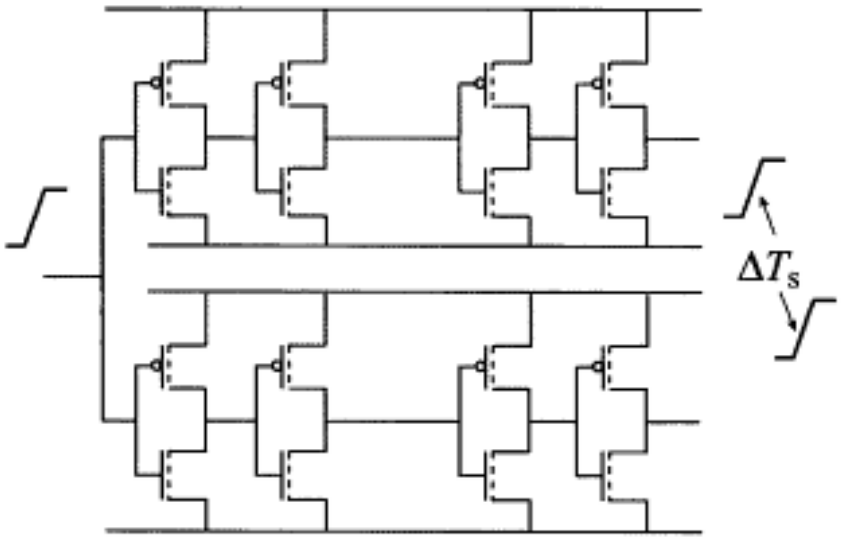

Figure 2.5. Jitter due to process, supply voltage and temperature variations (PVT)

As seen from the jitter sources, there are both environmental and static variations that contribute to the jitter performance. Therefore jitter is categorized in two components [13] which the total jitter is composed of (figure 2.6):

- Random jitter (Unbound, environmental variations)
- Deterministic jitter (Bound, process variations, layout)

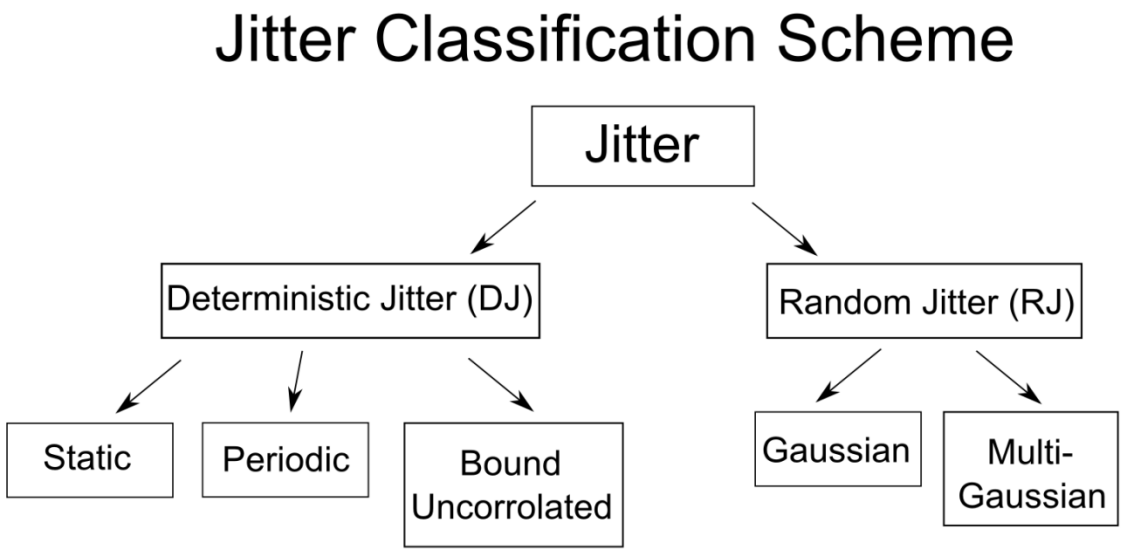

Figure 2.6. Jitter Classification Scheme [14]

Random jitter is theoretically unbound and has a Gaussian in distribution (figure 2.2) [15]. It is believed to follow this distribution due to the fact that one of the main sources of random jitter is thermal noise which has a Gaussian distribution. The fact that it is unbound means that it is unlimited peak-to-peak and usually measured in terms of a root mean square (RMS) value. As the name applies, random jitter is caused by random environmental variations as mentioned previously, making it hard to predict or reproduce.

Deterministic jitter is bounded and it is predictable [16] as well as reproducible. Deterministic jitter is caused by system mechanisms such as coupling noise, process variations, impedance mismatch etc. [13].

As the total jitter consists of random jitter and deterministic jitter, it can be expressed by adding them together, but there are different approaches for correlated and uncorrelated noise sources [17] [18].

Equation 2.4 describes total jitter for correlated noise sources while equation 2.5 describes total jitter for uncorrelated noise sources:

$$T_j = RJ_{rms} + DJ_{ptp} \tag{2.4}$$

$$T_j = \sqrt{(RJ_{rms})^2 + \left(DJ_{ptp}\right)^2} \tag{2.5}$$

where RJ_{rms} is the random jitter RMS value and DJ_{ptp} is the deterministic jitter peak-to-peak value.

In a delay line there are usually several cascaded delay elements. The noise of the first delay element cell will be amplified by the second cell's transconductance, filtered out by the output capacitance and summed up with its own intrinsic noise. The jitter amplification parameter g_{mj} is therefore dependent on the output capacitance, the inverters transconductance and the noise frequency as shown in equation 2.6 [19]:

$$\overline{\Delta v_{Out}^2} = \overline{\Delta v_{In}^2} \cdot \left(\frac{g_m}{\omega C_L}\right)^2 = \overline{\Delta v_{In}^2} \cdot g_{mj} \tag{2.6}$$

where g_m is the cells transconductance, ω is the angular frequency ($2\pi f$) and C_L is the output load capacitance.

So why is it so important to test for jitter? As mentioned, increasing clock frequencies means higher sensibility for jitter, but depending on the application there are several other reasons to test for jitter [20]:

- Studies show a clear link between jitter and overall device function and bit error rate (BER)
- For serial communications device manufacturers (SCDM), jitter testing is less time consuming and more conclusive than for instance production BER testing
- Jitter is the final frontier for high speed logic
- Jitter translates to overall system performance

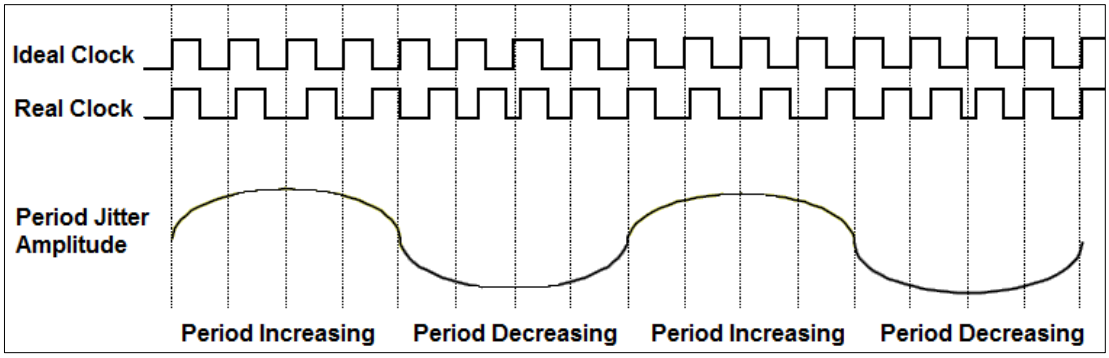

Figure 2.7. Real clock with periodical jitter versus ideal clock [21]

Figure 2.7 is an example of periodic clock jitter and is a main issue for high speed logic whereas the clock timing is essential. The periodic clock jitter obviously limits the resolution of the delay lines. A useful tool for examining jitter in a digital system is the eye-diagram, depicted in figure 2.8.

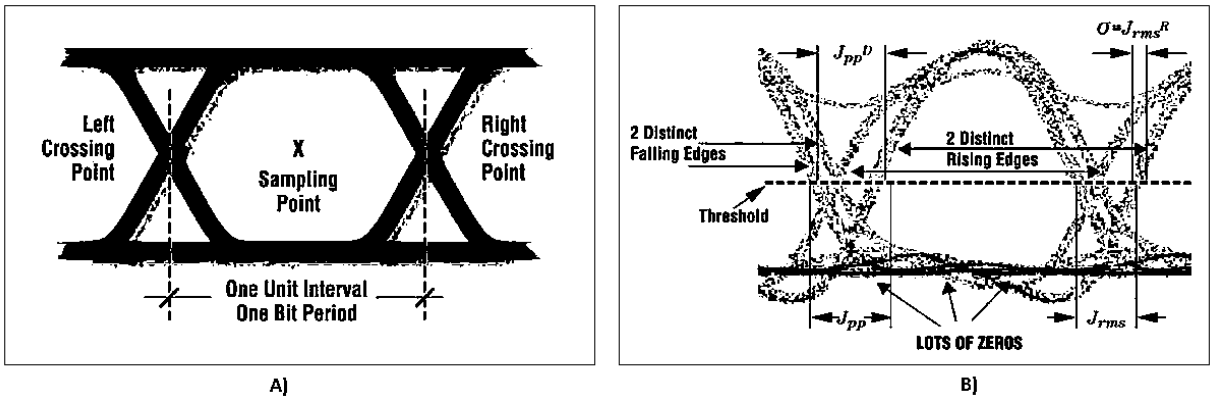

Figure 2.8. Idealized eye diagram (A) versus irregularly shaped eye diagram (B) [22]

One of the main reasons for using the eye diagram is its versatility as several measurements can be obtained from analyzing it (figure 2.9) [23], such as:

1. Zero Level
2. One Level
3. Rise Time
4. Fall time
5. Eye Height
6. Eye Width
7. Deterministic Jitter
8. Eye Amplitude
9. Bit Rate

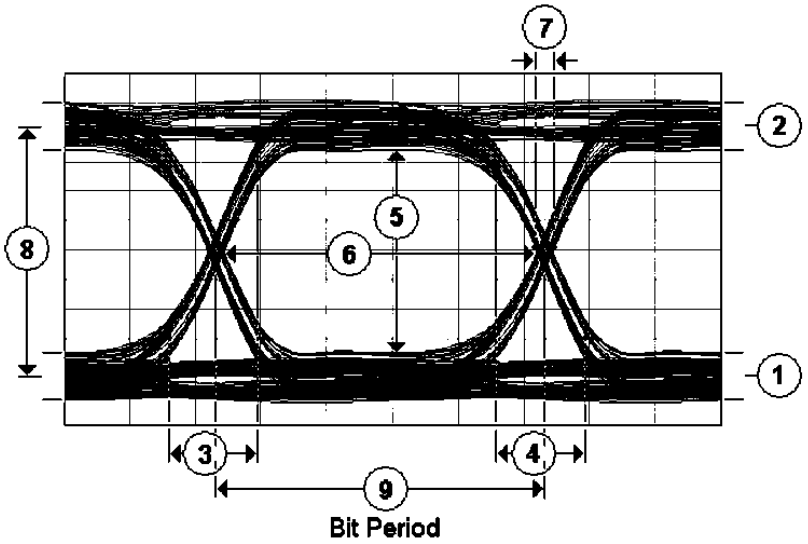

Figure 2.9. Different measurements obtained from the eye diagram [23]

2.2.2 Thermal Noise

Thermal noise (Nyquist noise or Johnson noise) is caused by the random thermally excited vibration of the charge carriers in a conductor and is the most often encountered noise source [24]. It is also a main source for random jitter [25]. The random thermally excited charge carriers create instantaneous current fluctuations which give rise to a noise voltage across the terminals of the conductor. A higher temperature will result in increased excitation of the charge carriers, thus a higher noise voltage. Thermal noise is also dependent on the resistive value of the device as expressed in equation 2.7 [26]:

$$E_t = \sqrt{4kTR\Delta f} \tag{2.7}$$

where k is Boltzmann's constant, T is the temperature in kelvin, R is the resistance or real part of the conductor's impedance and Δf is the noise bandwidth.

From equation 2.7 it seems that an open circuit with infinitely high resistance will generate an infinitely high noise voltage, but this is not the case since there is always a parasitic capacitance between the terminals creating a low-pass filter. This result in the kT/C limitation described by equation 2.8 [27] and displayed in figure 2.10. The kT/C limitation can be utilized constructively to reduce the output noise voltage by inserting a shunt capacitor.

$$E_n = \sqrt{\frac{kT}{C}} \tag{2.8}$$

where k is Boltzmann's constant, T is the temperature in kelvin and C is the capacitance in farads.

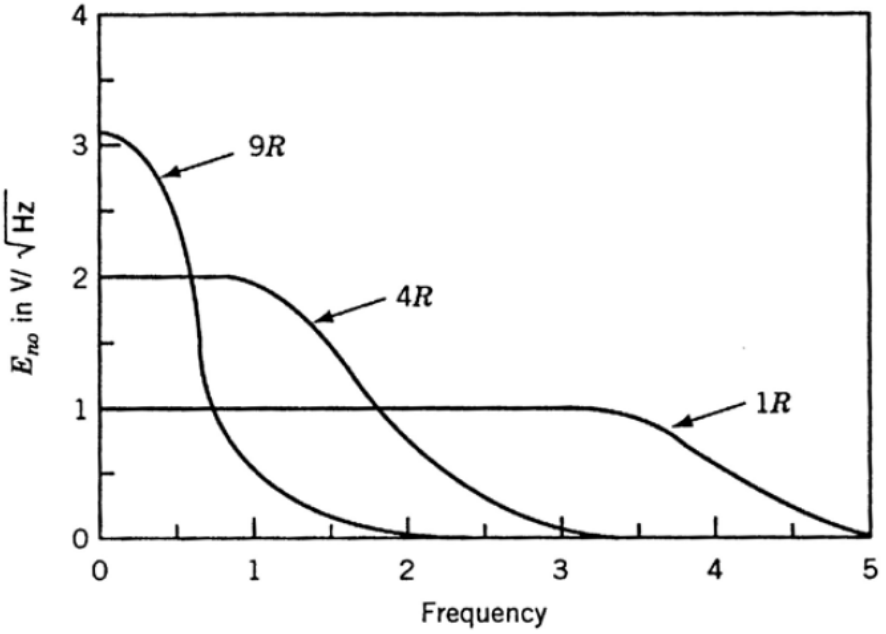

Figure 2.10. Noise spectral density for a resistance shunted by a capacitance [27]

As an example using equation 2.7, the voltage noise produced by a $1k\Omega$ resistor in a noise bandwidth of $1Hz$ is $4nV\ RMS$. This is often used as a reference level as it can easily be scaled up or down with the square root of the resistance and/or bandwidth.

Jitter induced by thermal noise is not dependent on the output load capacitance compared to the flicker noise induced jitter [28].

2.2.3 Flicker Noise

Flicker noise occurs in the transition between a crystal structure and another non-crystal structure, meaning that it is a common noise source in semiconductors [29]. In a $MOSFET$ device this transition occurs between the gate insulation (Silicon Oxide) and the substrate. In these transitions some loosely connected electron pairs will occur and these will be able to collect charges for some time as illustrated in figure 2.11:

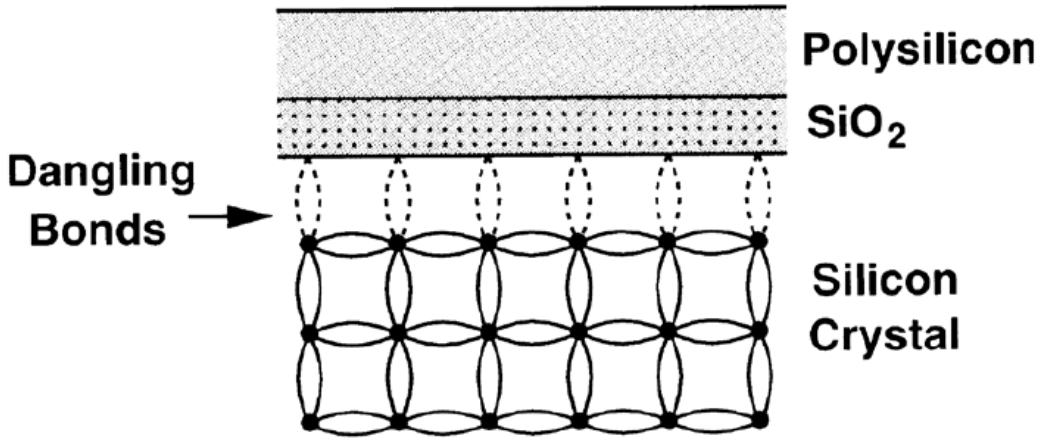

Figure 2.11. Transition between crystal and non-crystal structure in a MOSFET device

Flicker noise has a $1/f$ characteristic meaning that it decreases with frequency. Figure 2.12 is a plot from a simulation program with integrated circuit emphasis ($SPICE$), displaying the typical $1/f$ characteristic for an open loop $OPAMP$ configuration. Here it is clear that the output noise is dominated by flicker noise up to around $1kHz$, whereas the shot noise and thermal noise determines the noise level limitations at higher frequencies (figure 2.13).

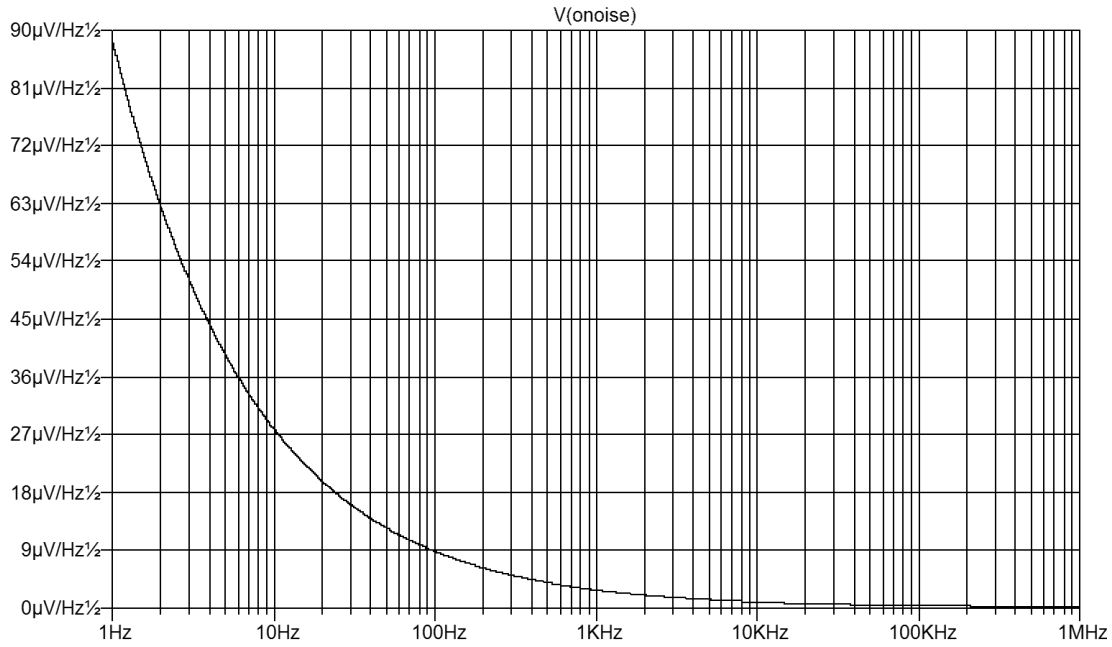

Figure 2.12. SPICE Simulation of output noise for an open loop OPAMP configuration

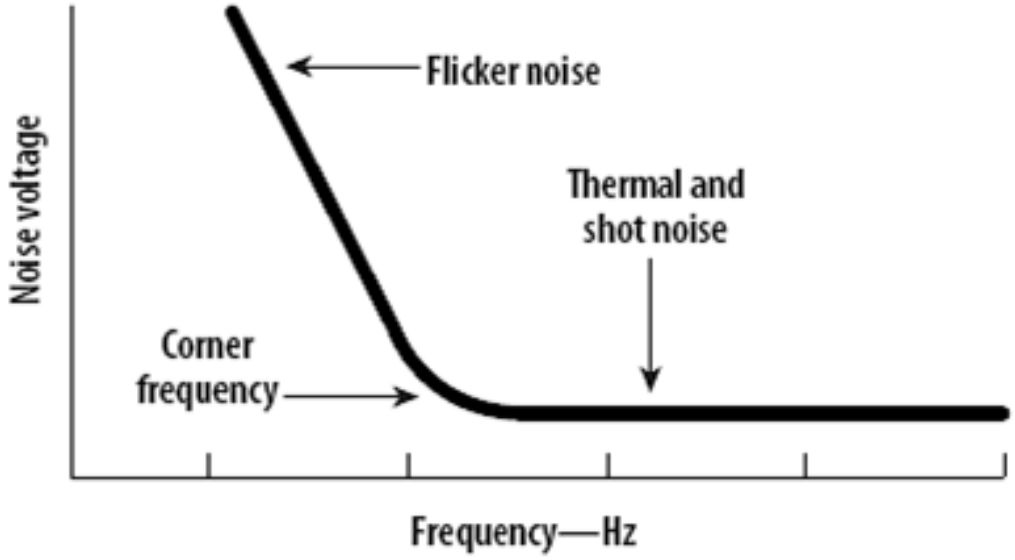

Figure 2.13. Identifying noise sources on a noise plot [30]

The magnitude of the flicker noise is dependent on the drain current, the oxide capacitance, effective channel length of the transistor and a process parameter as shown in equation 2.9 [31]:

$$I_f = \frac{1}{f} \cdot \frac{I_{DQ}^{A_F} \cdot K_F}{C_{ox} \cdot {L_{eff}}^2} \quad (2.9)$$

where K_F is the flicker noise coefficient, I_{DQ} is the quiescent current, A_F is a process dependent constant, f is the frequency in hertz, C_{ox} is the gate oxide capacitance and L_{eff} is the effective cannel length.

The flicker noise is mainly present in semiconductors due to the crystal structure but also exists in a smaller degree in resistances and other components such as radio tubes [29]. The flicker noise induced jitter is highly dependent on the output capacitance and also on the short-circuit current in the charge/discharge phase [19].

2.2.4 Shot Noise

Shot noise is a relatively small noise source and is caused by the PN (diode) transition in the transistor. The current flowing between the drain and source of the transistor randomly fluctuates, meaning that the shot noise has more impact as the currents are decreasing with transistor scaling. An intuitive comparison can be a coin toss, where the probabilities of the two outcomes are calculated. If the coin is tossed a large number of times, the probability of heads or tails will converge towards 50%. On the other hand, if the coin is tossed a few numbers of times the result may differ vastly from 50%. The random fluctuations therefore decrease in significance relative to the number of coins tossed, or in this case the number of electrons creating the current as shown in equation 2.10 [32] and figure 2.14:

$$I_{shot} = \sqrt{2 \cdot q \cdot I_{DC} \cdot \Delta f} \quad (2.10)$$

where q is the elementary charge or the electric charge carried by a single electron ($1.602\ x\ 10^{-19}$ Coulombs), I_{DC} is the direct current in amperes and Δf is the noise bandwidth in hertz.

Shot noise is usually an insignificant noise source due to the fact that these random fluctuations are minimalistic compared to the current itself and other noise sources, even though it becomes more significant with scaling. As an example, a current of $50mA$ consists of $3.12\ x\ 10^{17}$ electrons per second, so quite many variations can be tolerated before it becomes an issue. However, shot noise is temperature and frequency independent (eq. 2.9) in contrast to thermal noise and flicker noise. This means that at high frequencies and low temperatures shot noise may become the dominant noise source.

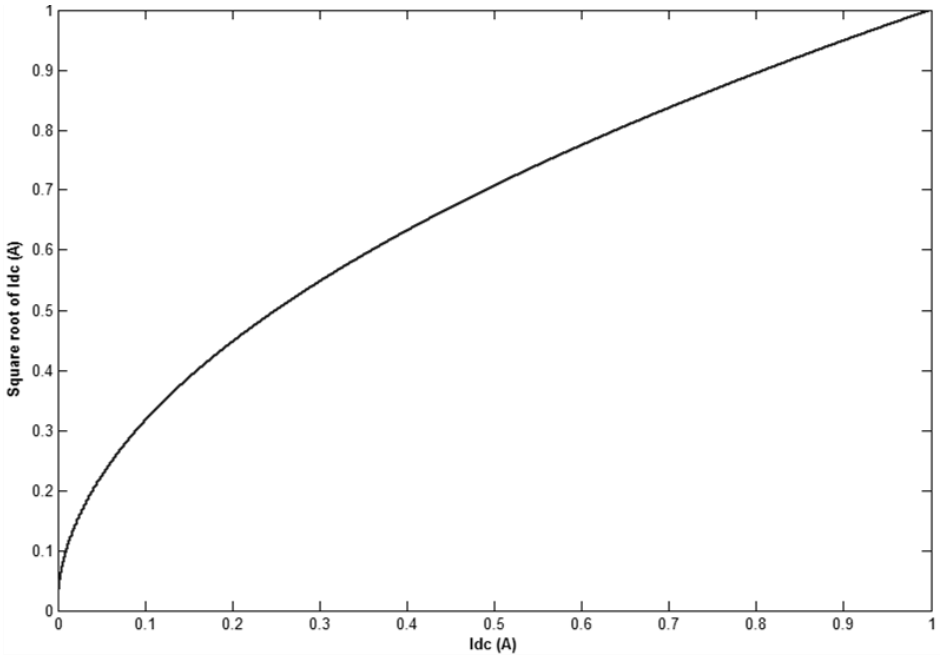

Figure 2.14. Square root of I_{DC} function

2.2.5 Coupling Noise

There are two types of electronic noise, the component noise as discussed previously and coupling noise. Coupling noise or electromagnetic interference (EMI) consists of any unwanted disturbance of electrical origin either conducted or radiated from an external source that effects and degrades the system performance. EMI is directly related to the system layout and noise level [33]. Coupling noise is caused by the interconnection of mutual conductors on chip as the density of electronic packaging continues to increase and equally important; the operating frequencies are increasing. Electric or magnetic fields from neighboring circuits or radiation from other sources will affect the noise and performance of a circuit.

A $90nm$ low power $CMOS$ process, as used in this thesis, offers high performance but parasitic effects are becoming increasingly important. These parasites, both capacitive and inductive, will transfer noise between nodes and result in inferior circuit performance. Examples of coupling noise sources are [34]:

- Mutual impedance
- Parasitic capacitance
- Parasitic inductance
- Parasitic resistance
- Capacitive coupling
- Inductive coupling
- Electromagnetic Radiation

As current flows in a conductor it creates an electric and magnetic field surrounding the conductor (figure 2.15). These radiated electromagnetic fields may interrupt other parts of the circuit or other circuits. If a conductor is placed within the proximity of such an electromagnetic field it will induce a noise voltage or a noise current depending on the radiated field since all conductors have parasitic capacitance and inductance.

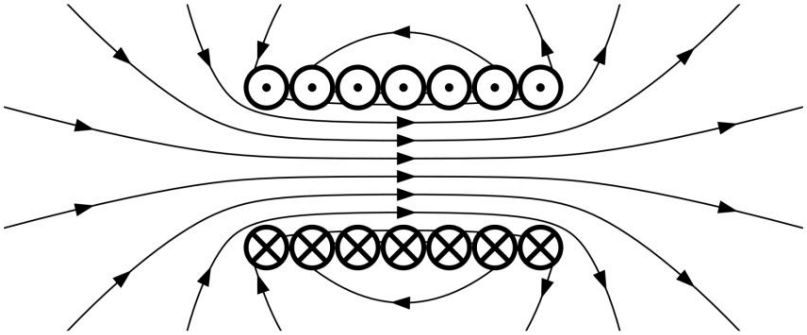

Figure 2.15. Magnetic field surrounding a current carrying conductor [35]

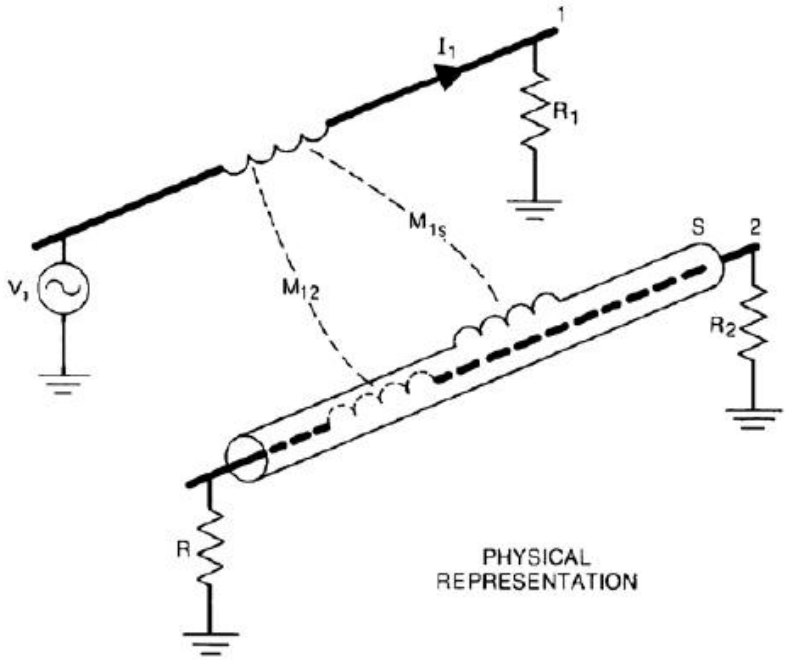

Figure 2.16. Physical representation of magnetic coupling [36]

The magnetic field induced noise voltage in a conductor using Biot-Savarts law can be expressed as (figure 2.16) [36]:

$$V_n = j\omega M_{12} I_1 \tag{2.11}$$

where j is the imaginary unit ($\sqrt{-1}$), ω is the angular frequency ($2\pi f$), M_{12} is the mutual inductance between inductor 1 and conductor 2 and I_1 is the current in conductor 1.

As seen from equation 2.11 the noise voltage is an important consideration in high speed electronics as it increases with frequency and current. The mutual inductance between the two conductors is dependent on the distance between the conductors.

The best way to reduce the magnetically induced noise voltage from a radiation source is to reduce the loop area of the receptor (figure 2.17) [37]. Other techniques to reduce magnetic coupling noise can be:

- Increase the distance to the noise source
- Decrease the frequency
- Reduce the mutual inductance
- Reduce source current
- Shielding (Coax, STP, FTP cables) (figure 2.18)
- Twisting the source conductors (TP, FTP and STP cables)
- Twisting receptor conductors (TP, FTP and STP cables)
- Normal orientation of source and receptor
- Proper termination
- Balanced signal reference

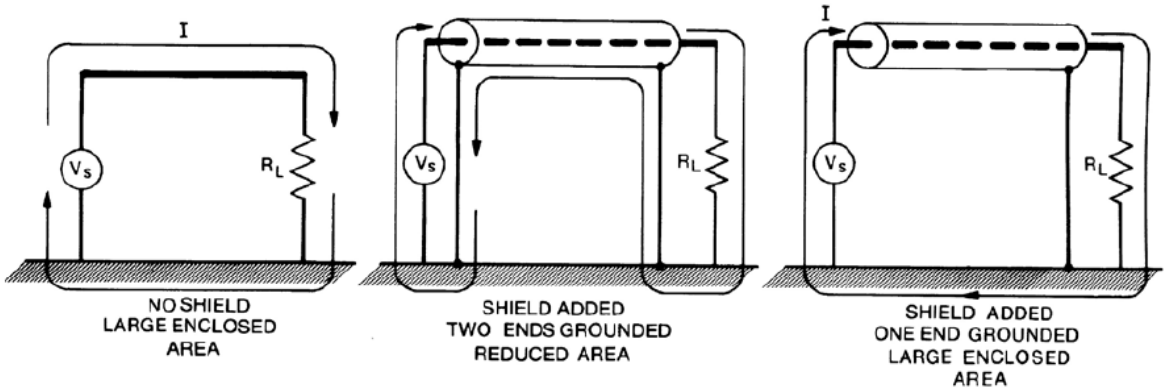

Figure 2.17. Effect of receptor loop area reduction [37]

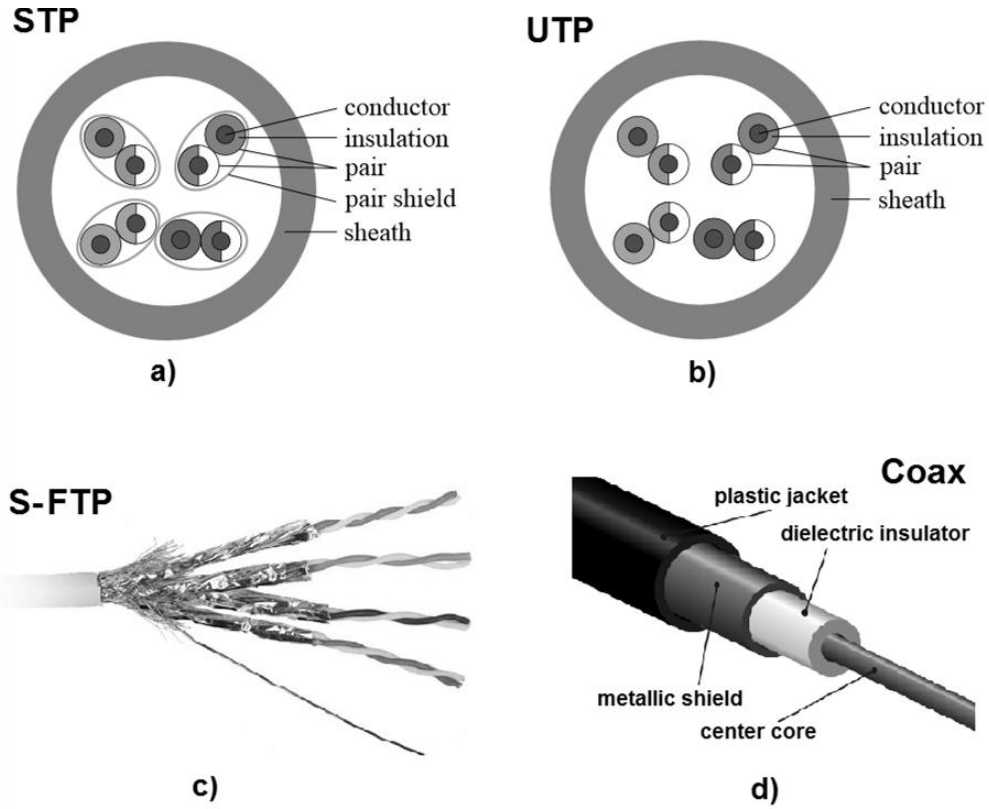

Figure 2.18. STP, UTP, S-FTP and Coax cables used to reduce EMI

It is to be noted that a shield does not actually work as a shield, blocking the magnetic field out as it does with electric fields. Instead it absorbs as much as possible of the noise so that the noise voltage in both shield and conductor differentially cancel each other out. If the current passes through the shield instead of the ground plane in the opposite way of the conductor inside, the effective loop area is also reduced. If the current passing through the shield is equal to but opposite of the conductor inside the magnetic field outside of the cable is eliminated. This reduces the radiation to other parts of the circuit. Even though shielding is not implemented in the design it is important theory that supplies understanding of how noise is transmitted in electronic circuits and the techniques that can be utilized to reduce the effect of it.

The electric field induced noise between two conductors (figure 2.19) can be expressed by equation 2.12 [38]:

$$V_n = \frac{X_{C_{2G}} || R}{\left(X_{C_{2G}} || R\right) + X_{C_{12}}} V_1 \tag{2.12}$$

where $X_{C_{2G}}$ is the impedance of capacitor C_{2G}, R is the resistance of the conductor, $X_{C_{12}}$ is the impedance of capacitor C_{12}, V_1 is the source conductor voltage and $X_{C_{2G}} || R$ is the parallel impedance of $X_{C_{2G}}$ and R.

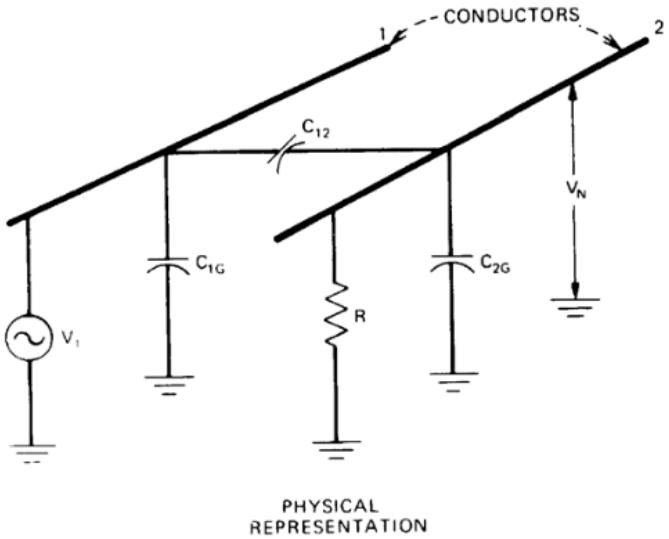

Figure 2.19. Physical representation of electric coupling [38]

To reduce the electrical field induced noise many of the techniques applied to reduce magnetic field noise can be used in addition to various techniques specialized to reduce the electrical field noise. Some of these techniques are [39]:

- Generate counter noise
- Avoid crossing of cables/routings
- Minimize conductor width at crossing section
- Increase distance between conductors
- Shielding
- Capacitive ground load
- Use isolation with lower dielectric constant ε
- Reduce output resistance
- Decrease the frequency
- Reduce the voltage swing

In an application specific integrated circuit $(ASIC)$ some of the mentioned techniques cannot be used as it is not possible to twist routings etc. on-chip, but there are existing methods for implementing shielding on an $ASIC$ [40] as illustrated in figure 2.20. Other effective reduction techniques on an $ASIC$ is to add the noise source in the highest metal layer, use ground plane, separating supply lines for analog and digital logic and to add guard rings.

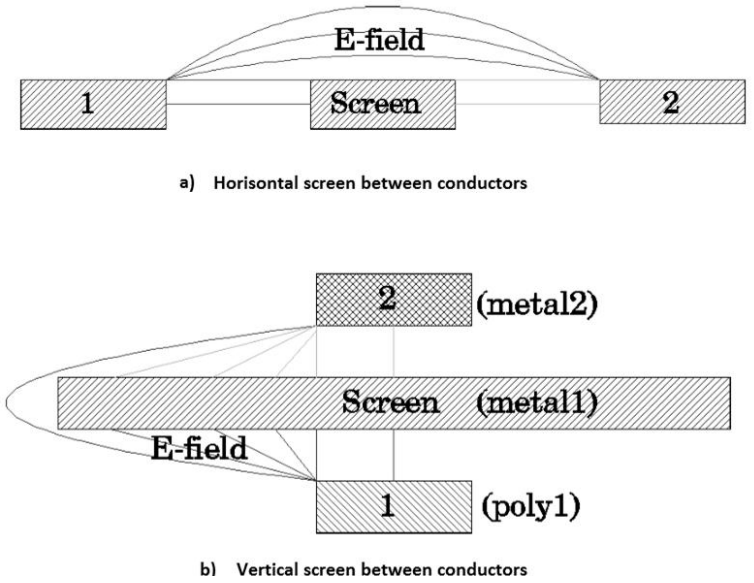

Figure 2.20. Horizontal and vertical shielding implemented on an ASIC

In the design some of the techniques to reduce the magnetic and electric field induced noise were implemented. For instance the distances between the conductors were kept above the minimum technology distance, the crossings of routings were kept to a minimum and long routing loops were avoided.

Other radiation sources such as cosmic radiation may also affect some sensitive circuits but is not covered here as it has a very low influence. For electronics used in space or in harsh environments (RAD-Hard circuits) this becomes more important.

Compared to component noise, coupling noise can be reduced or eliminated by the designer, but it requires skill, time and effort.

2.2.6 Temperature Variations

Temperature variations are caused by temperature as it provides kinetic energy to the atoms, making them vibrate. Temperature actually *is* a measure of kinetic energy. In addition to the thermal noise, temperature variations also affect other device parameters. A higher temperature will increase the kinetic energy of the atoms meaning that the bound electrons will move more rapidly. This increases the threshold voltage V_t and reduces the current. More specifically this increases the off-current I_{off} and reduces the on-current I_{on} [41] (figure 2.21) which is the worst case regarding device performance. The cell becomes slower and uses more power in off mode due to increased leakage. The off-current is increased because additional loosely bound electrons are released in to the substrate due to the added thermal energy, making a channel for current to flow.

The on-current is reduced as the higher excited electrons create additional resistance for the free electrons, thus lowering the mobility of the free electrons. A lower temperature has the opposite effect.

It is to be noted that temperature variations are caused not only by the ambient temperature but also by the device itself as it generates heat. For instance the delay through the delay line will be shorter in the beginning when the circuit is “cold” and increase as the internal temperature rises and settle at a certain level depending on the activity factor α. The power consumption is an important factor as higher power consumption will generate more heat. The dynamic power consumption can be expressed by equation 2.13 [42]. Power density of state-of the art technologies is extremely high, approaching $30W/cm^2$ in Pentium 3 processors [42]. As a comparison this is around 3 times the power density of a hot plate.

$$P_{dynamic} = \alpha C_L V_O^2 f \tag{2.13}$$

where α is the activity factor, C_L is the load capacitance, $V_O$is the output voltage swing and f is the frequency.

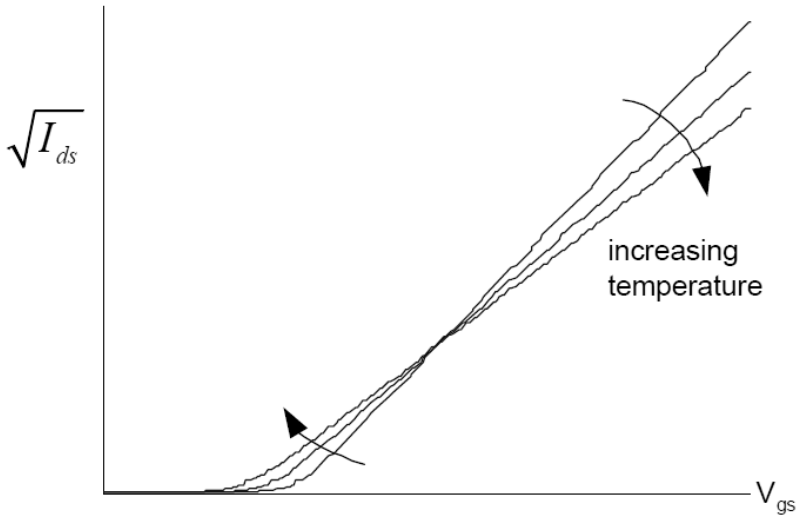

Figure 2.21. Temperature sensitivity for MOSFET’s [41]

If the temperature is high enough (~150˚C) the covalent structure of the semiconductor will break down and the device will stop functioning.

One way of reducing the dependency on temperature is to use dynamic power compensation [43]. This compensates for the drastic change in power consumption from static to dynamic states in $CMOS$ logic, thus reducing the delay variations. Such vast difference in power consumption relates to fact that charging and discharging of capacitors at high frequencies require much more power than the standby leakage.

The dynamic power consumption will ensure a more constant power consumption, thus also a more constant temperature for all operations.

2.3 Power Supply Variations

Power supply noise is an important noise factor in modern design as they can cause unwanted noise and oscillations. As the power supply has a finite bandwidth the output impedance increases with frequency resulting in a noise voltage [44]. This can be modeled as a series inductance with the output. To reduce the noise, the inductance or the rate of change of the current has to be reduced. The inductance can be reduced by decreasing the wire's length or use multiple isolated wires. There are mainly two ways of minimizing noise, each for its own purpose, is bypassing and decoupling (figure 2.22).

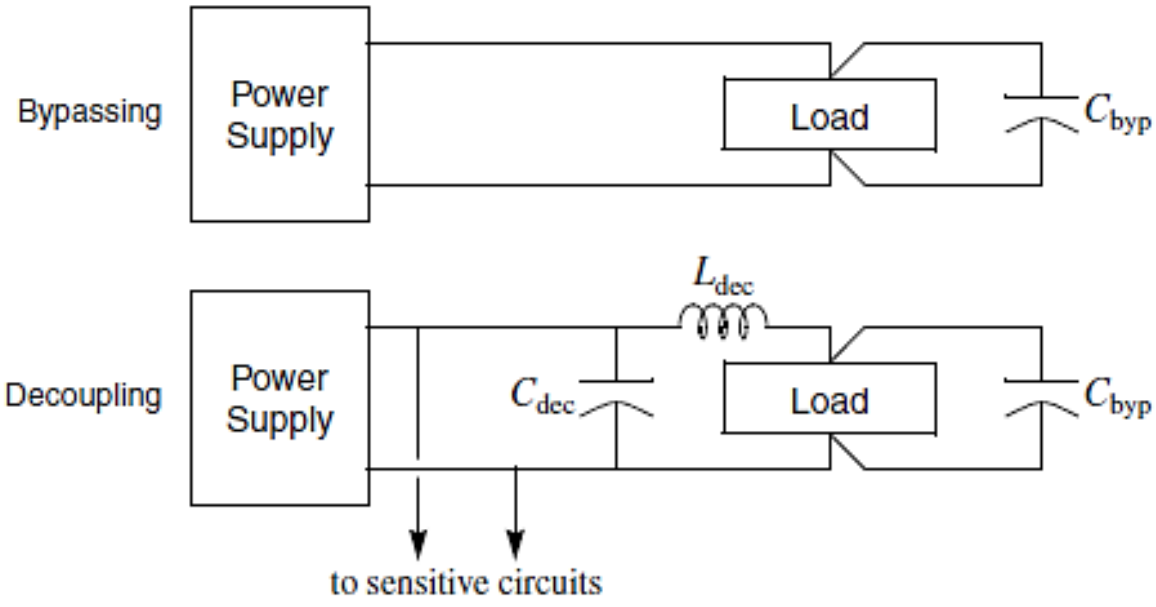

Figure 2.22. Bypassing and Decoupling [45]

Adding a shunt capacitor will create a low impedance path for high frequencies as well as an energy storage that can respond much quicker than the power supply. The capacitor will respond to rapid current changes and try to maintain the desired voltage level. It also acts like a low-pass filter filtering out the high frequency noise.

To optimize the effect of the bypassing capacitor, since all capacitors have parasitic series inductance, the capacitor with the highest capacitance in the smallest sized package should be used. It is always better to use several capacitors in parallel [44] to reduce the inductance further. Placing the bypassing capacitor(s) near the switching device is preferred as this will increase the bypassing effect [46]. The use of bypassing capacitors is becoming increasingly important as the clock frequencies increase and the power supply limitations become imminent.

Decoupling is used to isolate two circuits on a common supply line [45]. The decoupling network is usually a low-pass filter with a high impedance element in series with the supply line. This might be a resistor if a direct current (DC) voltage drop is not an issue. In figure 2.22 the bypass capacitor is also included as bypassing is always used in practice when decoupling.

In the design of the delay lines bypassing was implemented using bypass capacitors as well as a grid of VDD and GND supply lines to further reduce the supply noise.

2.4 Layout Issues

Layout issues, in addition to coupling noise, such as non-uniform routing can contribute to different propagation delay as seen in figure 2.23. This is especially important in high resolution applications.

The major deviations in figure 2.23 are due to non-uniform routing while the smaller groupings of deviations are due to mismatch and process variations. Non-uniform routing contributes to different signal path lengths, thus different time constants given by equation 2.14:

$$\tau = RC \tag{2.14}$$

where R is the resistance in ohms and C is the capacitance in farads.

From equation 2.14 it is easy to see that a longer signal path, which has higher resistance and capacitance, will have a higher time constant than a shorter signal path. Not only will the length of the signal path, also the number of via contacts used, affect the delay as they introduce higher resistive values. This means that if the routing is non-uniform the delays will differ from branch to branch as seen in figure 2.24. Here it is clear that the difference in delay between delay element 5 and 6 will be much greater than the rest which has an equally long signal path length between the delay elements.

Therefore it is extremely important to find the critical signal paths and design these uniformly in the layout if the delay elements shall exhibit the desired high resolution. It is the difference in delay between supposedly matched cells that is the main issue, not to achieve the shortest possible delay for all cells, even though this might be desirable. The intrinsic delay can be removed by adding it to both the transmitter and receiver.

Compared to mismatch and process variations, layout issues can be reduced or eliminated by the designer, but it requires skill, time and effort. Some of the layout techniques used to achieve high system performance, many of which is implemented in the design, is discussed in chapter 4.

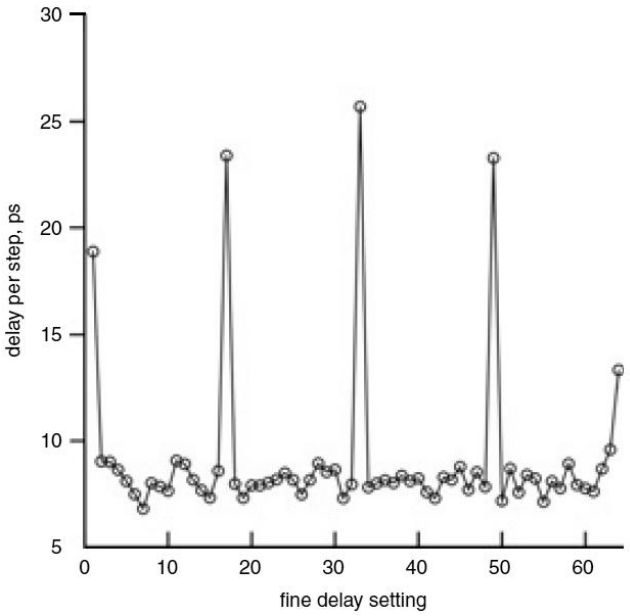

Figure 2.23. Effect of non-uniform routing [47]

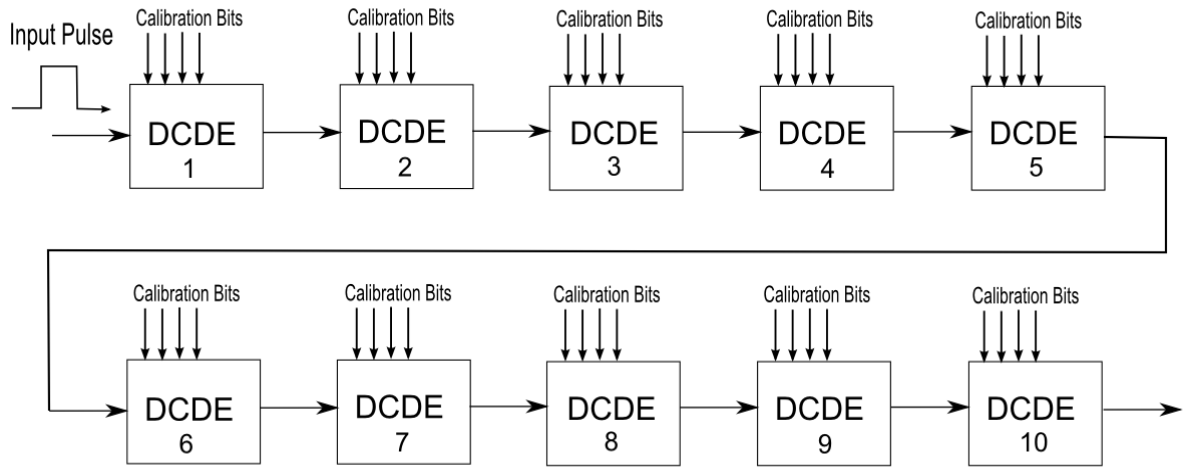

Figure 2.24. Example of S-shaped distribution of delay elements contributing to non-uniform routing

2.5 PRF

PRF is a term used in radar technology and as the name implies, the PRF is the number of pulses transmitted per unit of time. PRF can be used to determine the range of the radar, as the pulses can only cover a certain distance before the next pulse is being sent (figure 2.25). The maximum unambiguous range can be expressed as:

$$Range = \frac{v}{2 \cdot PRF} \tag{2.15}$$

where v is the speed of propagation, PRF is the pulse repetition frequency and the factor $1/2$ comes from the fact that the total distance the pulse has to travel is twice the distance to the object as shown in figure 2.25.

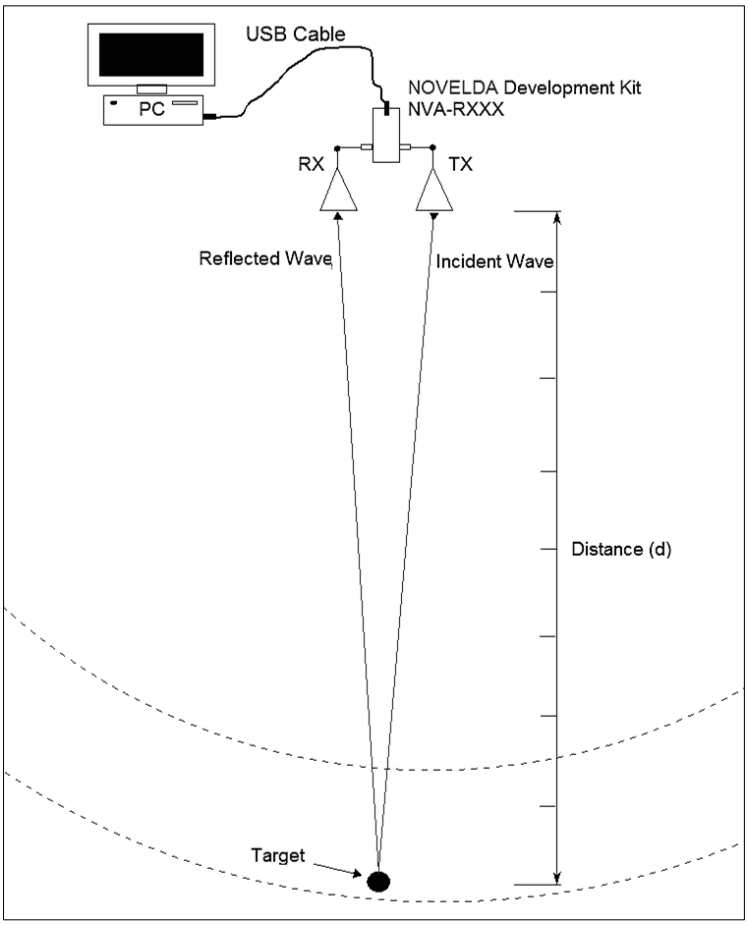

Figure 2.25. Total distance covered by emitted pulse

Since the noise in a circuit depends on the frequency as explained previously, the PRF will contribute to noise and jitter, especially if the PRF is high.

2.6 Slew Rate

A signals slew rate describes the rise or fall-time of the propagating signal and has an effect on the jitter noise due to the uncertainty of when the output of the element will switch. A given amount of noise voltage will produce a variance in the time domain given by the slew rate of the input signal [48] as expressed in equation 2.16:

$$\overline{\Delta t^2} = \overline{\Delta v^2} \cdot \left(\frac{C_L}{I_L}\right)^2 \tag{2.16}$$

where Δt is the change in time or jitter, Δv is the change in voltage, C_L is the load capacitance and I_L is the load current.

As seen from equation 2.16 the main factors limiting the slew rate is the current and load capacitance. The slew rate can be expressed by equation 2.17 and the different parameters are illustrated in figure 2.26. Slew rate is commonly expressed in $V/\mu s$.

$$SR = \frac{\Delta v}{\Delta t} = \frac{(V_{OH} - V_{OL}) \cdot 80\%}{t_r} \tag{2.17}$$

where V_{OH} is the outputs highest voltage (usually V_{DD}), V_{OL} is the outputs lowest voltage (usually V_{SS}) and t_r is the rise time.

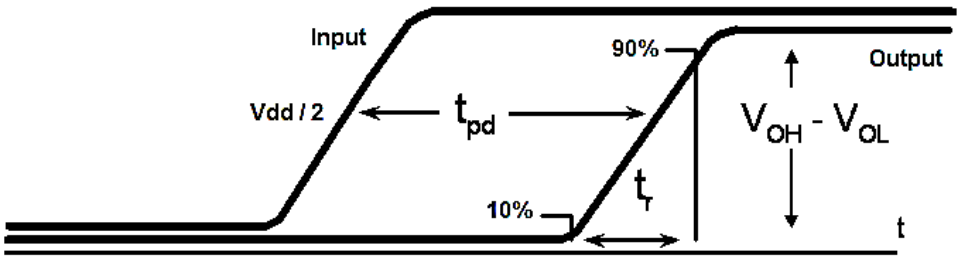

Figure 2.26. Illustration of slew rate parameters [49]

Viewing equations 2.16 and 2.17, it is clear that optimization of jitter performance it is utilized with a high as possible slew rate which implies fast switching delay elements. The higher frequency content of the steep pulses reduces the flicker noise which in return reduces the jitter.

Other than to minimize the noise voltage, the current should be increased and/or the load capacitance decreased to reduce the time variance. A lower slew rate on the input of an element will result in higher uncertainty of when the output will switch, while a faster slew rate will reduce the uncertainty.

2.7 Digitally Controllable Delay Element Architectures

To compensate for the environmental variations there are several types of delay elements that can be used, both analog and digitally controlled delay elements. Some of these architectures, with their own advantages and disadvantages, are listed in table 2.1. It is to be noted that there are many different architectures for each type of delay elements so the pros and cons listed in table 2.1 might not apply for all architectures in general.

Type of delay element	Advantages	Disadvantages
Current starved	➢ Simple structure ➢ Relatively wide range of delay tuning	➢ Non-linear transfer function ➢ Some architectures has different signal paths (figure 2.28)
Shunt capacitor	➢ Simple structure ➢ Fine delay resolution	➢ Larger area consumption ➢ Nonlinear transfer function (figure 2.29)
Variable resistor	➢ Simple structure for a small $n \times n$ matrix	➢ Non monotonic delay behavioral with ascending binary input ➢ Stacking of transistors ➢ Layout issues (routing) with large stack ➢ Variable signal path ➢ High temperature dependence

Table 2.1. Advantages and disadvantages of different types of digitally controllable delay elements [50].

The most common types of digitally controllable delay elements are:

- Current starved delay element (figure 2.27 and figure 2.28)
- Shunt capacitor delay element or load compensation (figure 2.29 and 2.31)
- Variable resistor delay element (figure 2.30)

To be able to fully control the propagation delay, a digitally controllable delay element is preferred to characterize and save the calibration data in RAM for later use. This allows us to create a complete "map" of all combinations and delays for different temperatures which can be used to ensure correct delay for almost all variations of operation. The most common problem with some existing $DCDE$ architectures is the non-monotonic delay behavioral with ascending binary input [51], the temperature dependency and the nonlinear transfer function as mentioned in table 2.1.

It is also to be noted that you get $n^2 - 1$ different delay configurations using n transistors in the variable resistor (figure 2.30) and current starved technique (figure 2.28) since one transistor always has to be on at the source of the inverter.

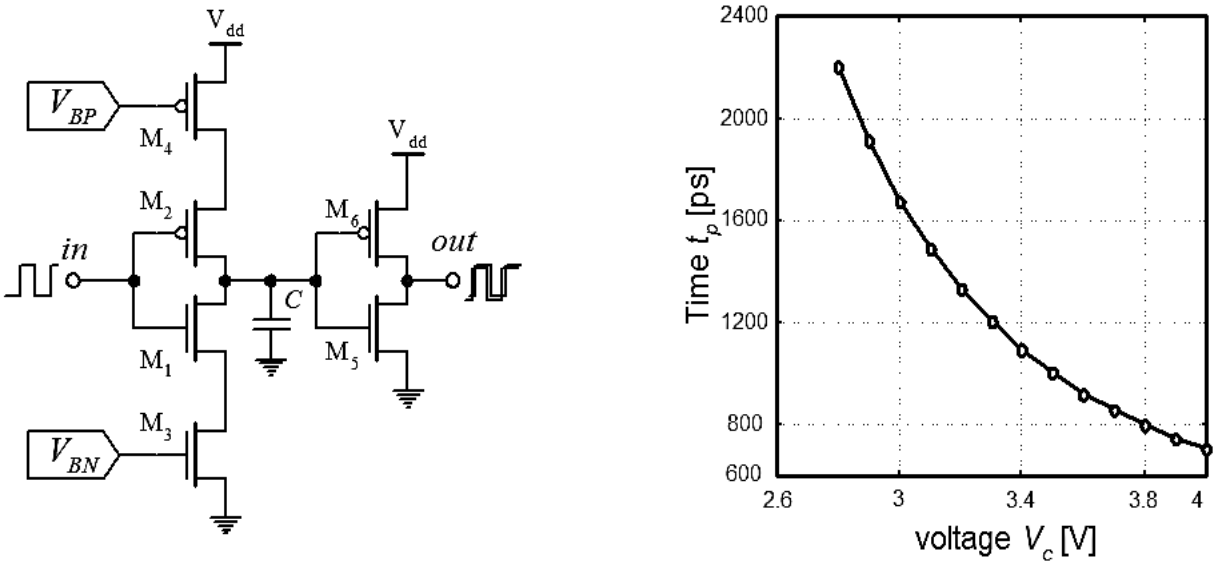

Figure 2.27. Analog current starved delay element (1.2µm $CMOS$ process) [50]

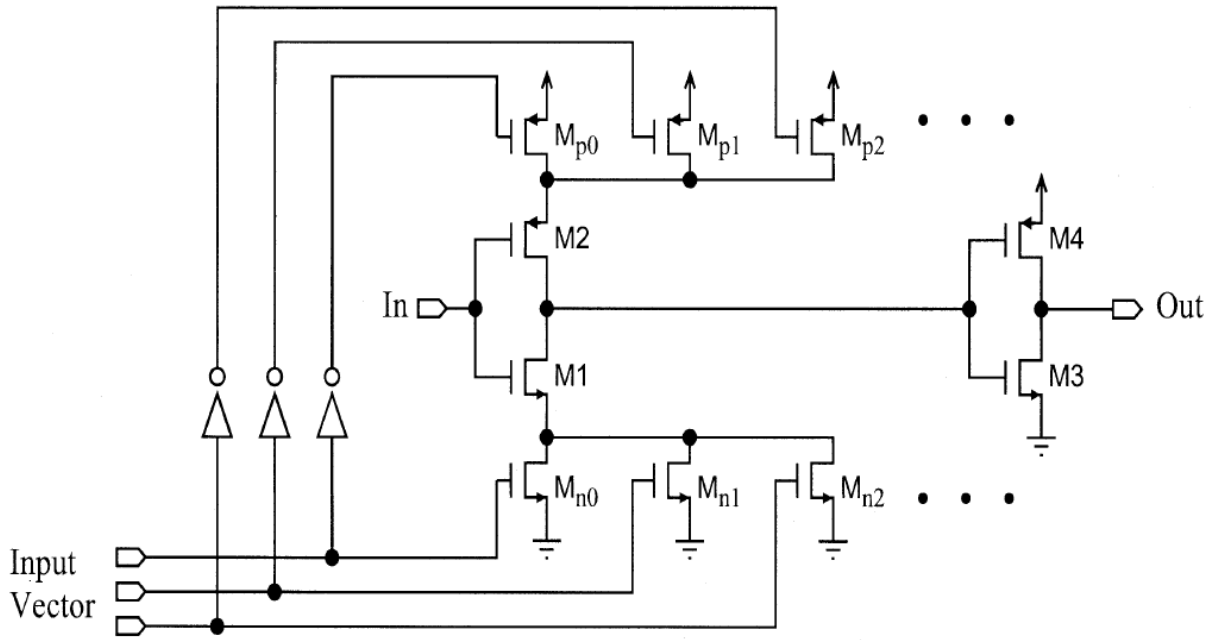

Figure 2.28. Digitally controlled current starved inverter element [51]

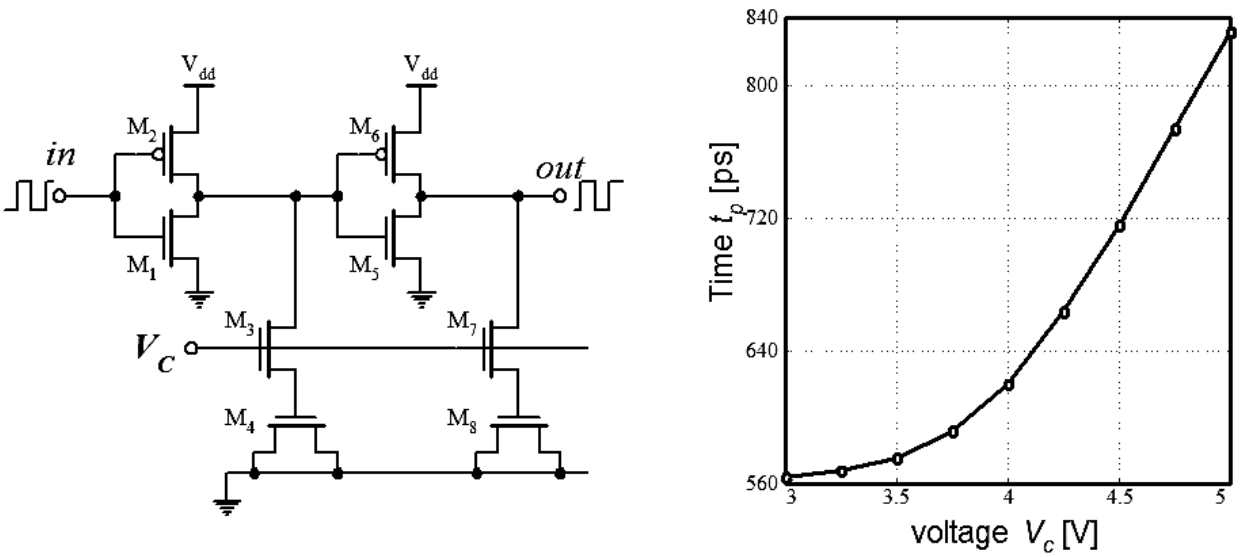

Figure 2.29. Analog shunt capacitor delay element (1.2µm CMOS process) [50]

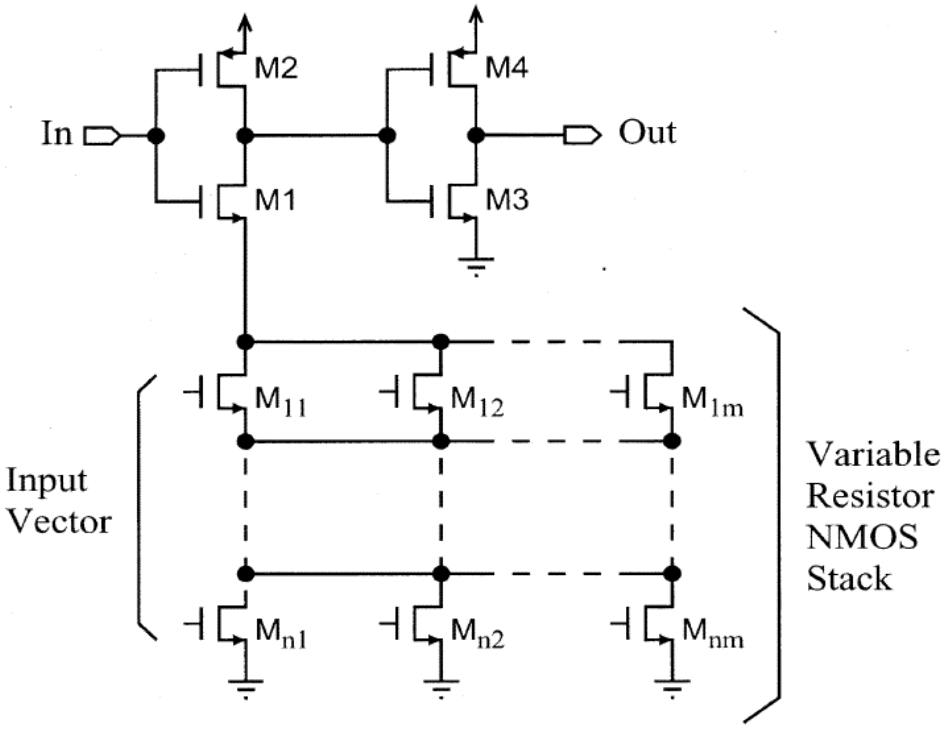

Figure 2.30. Digitally controlled variable resistor delay element [51]

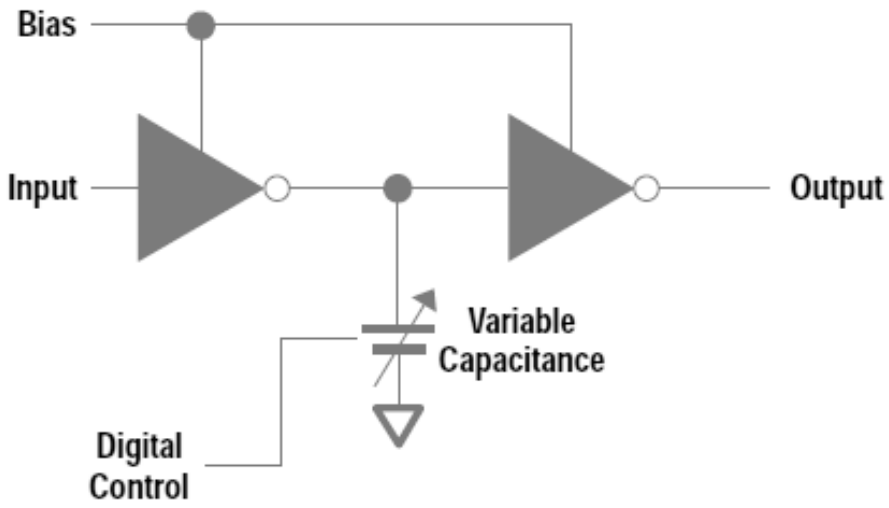

Figure 2.31. Digitally controllable shunt capacitor delay element block diagram [52]

The disadvantage of non-linear transfer function can be accounted for by using non-linear biasing, resulting in an overall linear transfer function [50].

Thermometer code can also be implemented [10] to reduce the glitching noise, therefore reduce the settling time and improve INL and DNL. A tradeoff with using thermometer code is the larger silicon area due to the fact that thermometer code uses $2^n - 1$ transistors for n bits, whereas the binary input only uses n transistors. Matching is also a problem as it contains more transistors. Figure 2.32 illustrates thermometer code versus binary and decimal-values.

	Binary			Thermometer Code						
Decimal	b_1	b_2	b_3	d_1	d_2	d_3	d_4	d_5	d_6	d_7
0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	1
2	0	1	0	0	0	0	0	0	1	1
3	0	1	1	0	0	0	0	1	1	1
4	1	0	0	0	0	0	1	1	1	1
5	1	0	1	0	0	1	1	1	1	1
6	1	1	0	0	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1

Figure 2.32. Thermometer code representation for 3-bit binary values [53]

2.8 Delay Lines

Several types of delay lines can be configured using the delay element architectures mentioned previously. Some types of delay lines consist of cascaded delay elements where the signal path is programmable [54], such as:

- Multiplexed delay lines (figure 2.33 and figure 2.34)
- Tapped delay lines (figure 2.33 and figure 2.36)
- Ramped comparators
- Charge-coupled devices
- ECL gate arrays

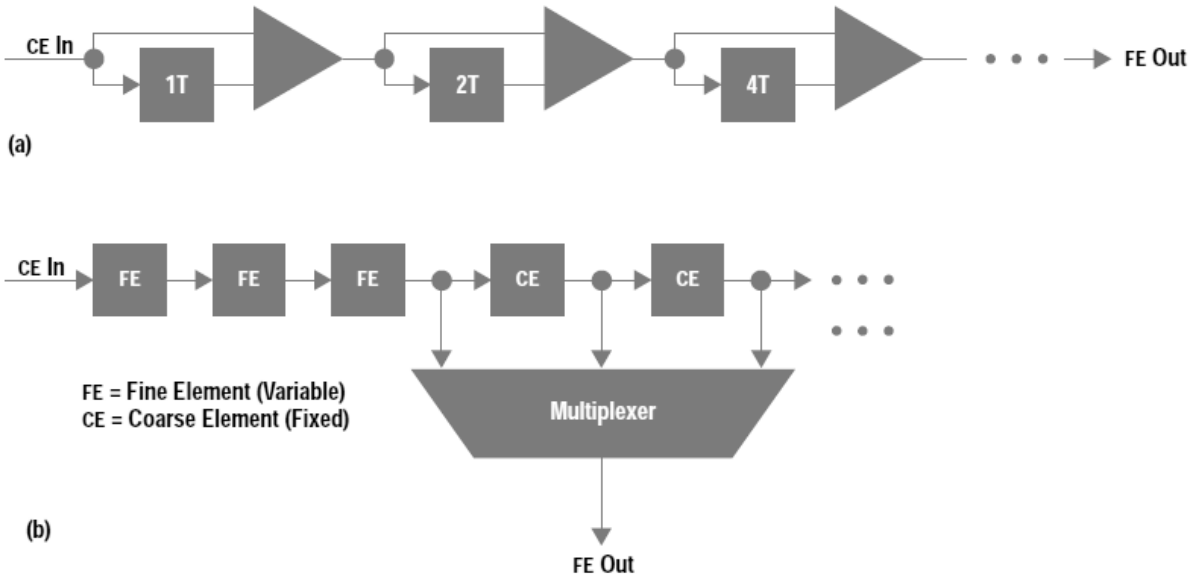

Figure 2.33. Multiplexed delay line (a) and tapped delay line (b) [54]

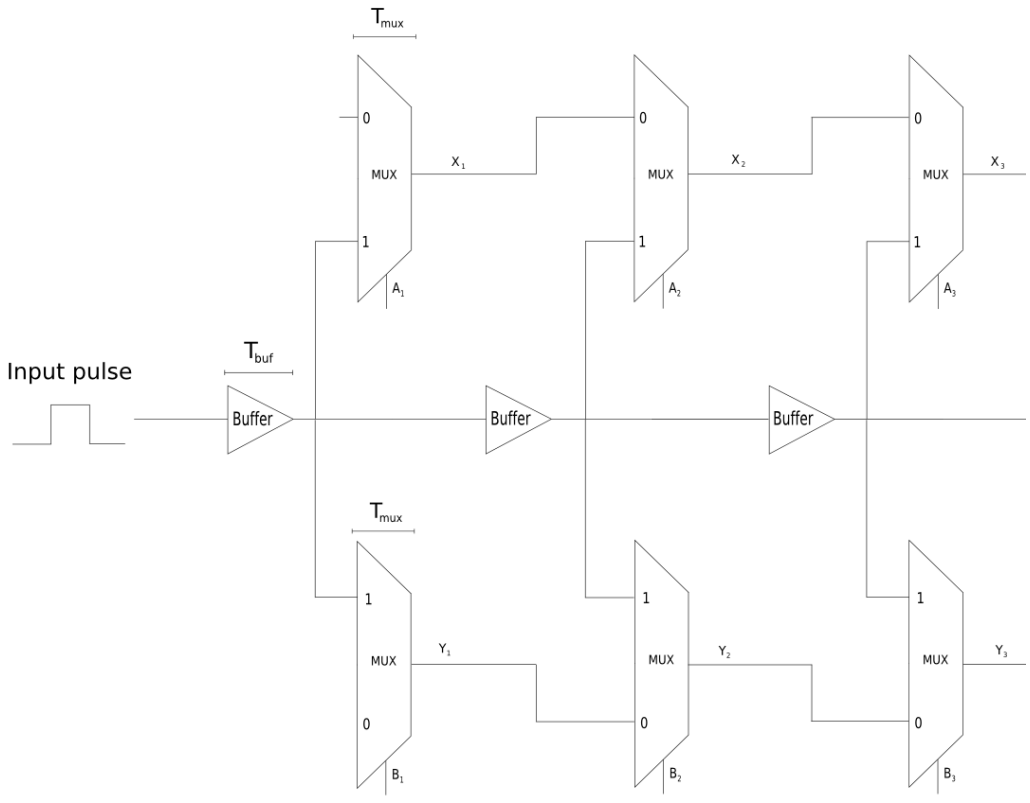

Figure 2.34. Differential binary weighted multiplexed delay line [55]

The differential multiplexed delay line shown in figure 2.34 uses the multiplexer select inputs A_n and B_n to determine the signal path and therefore the propagation delay of the signal. Since this is a differential delay line, the two outputs are subtracted from each other using digital logic to end up with a pulse with a width equivalent to the desired delay (figure 2.35).

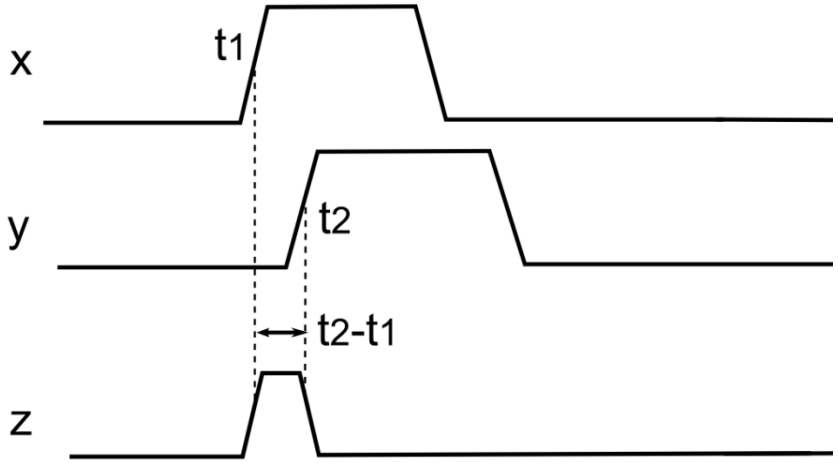

Figure 2.35. Logical operation $z = x\bar{y}$ for the differential multiplexed delay line [55]

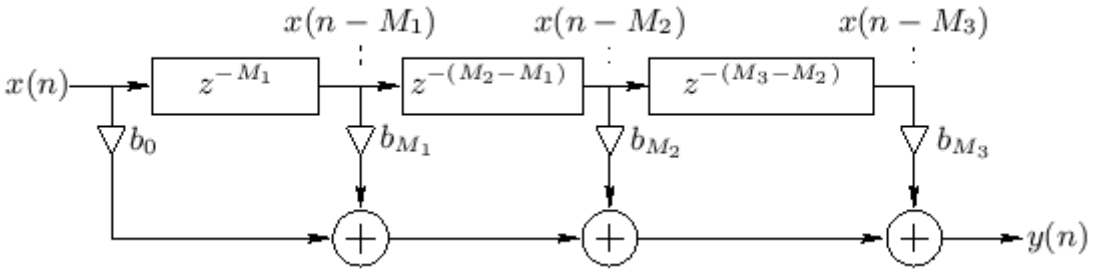

Figure 2.36. Mathematical representation of a tapped delay line [56]

The tapped delay line can exist of a self-composed mix of different delay elements as long as it has one or more tap where the delayed signal is extracted. Figure 2.36 depicts a mathematical representation of a tapped delay line with output function described in equation 2.18:

$$y(n) = b_0 x(n) + b_{M_1} x(n - M_1) + b_{M_2} x(n - M_2) + b_{M_3} x(n - M_3) \quad (2.18)$$

where $x(n)$ is the input signal, M_X is the intrinsic delay of delay element X and b_X is he scaling factor for tap X.

Combining these different delay elements, each with their own advantages and disadvantages, will give rise to several combinations of delay lines with different characteristics and performance to the problem of interest.

It is therefore possible to compensate for mismatch and non-uniformities by combining delay elements (and delay lines) of different propagation delay, effectively combining long and short delay elements for increased resolution (analogous to a Vernier scale). If each of these delay elements is measurable and tunable on-chip, the delay elements can be calibrated to have the desired propagation delay, confined by the resolution of the measurement result and the tunability of the delay elements. The problem statements of “How to design environmental and mismatch self-compensating, low jitter delay lines in $90nm$ low power $CMOS$?” might therefore be achieved using one or more of these techniques

3 Circuit Implementation

3.1 Delay Line Architectures

The first main thing to determine is how many delay lines, which types of delay elements to include within the delay lines and what kind of delay line structure to implement in the design. A decision of implementing 3 different delay lines became logical in relation to comparison purposes and a reasonable number of delay lines to design within the designated time period.

As the non-tunable buffer delay elements already were implemented in the current impulse radar system it became natural to use this as a basis for comparison. It is a simpler architecture and a benchmark for jitter performance. Furthermore the next choice of a current starved delay element was chosen due to the lower temperature dependency compared to other architectures [57]. The third choice of a shunt capacitor (load biasing) delay element was chosen to try out the use of $nMOS$ capacitors instead of traditional capacitors, as this reduces the silicon area, and see how it compared to the other architectures.

It also seemed logical, for comparison purposes, to use the same type of delay element architecture for each delay line in both coarse and medium tune, even though it would be possible to mix them. As a result the current starved delay line consists of only tunable current starved delay elements, the shunt capacitor delay line of only tunable shunt capacitor delay elements and the non-tunable buffer delay line of only non-tunable buffers.

Implementing a tapped delay line architecture was chosen for the high resolution and low jitter [54].

All the delay elements, both coarse tune and medium tune, are explained in greater detail in the succeeding chapters, layouts are added in appendix A and schematics are added in appendix B.

3.2 Specifications

One of the specifications for the circuit is that it should be optimized for a $100MHz$ PRF, meaning a main delay of $\frac{1}{100MHz} = 10ns$. The second and third specification was that the delay elements should be digitally controllable (tunable) and that the design should focus on low jitter respectively.

3.3 Top Level Schematics

The top level block diagram of the implemented multiplexed delay lines is shown in figure 3.1. The contents of the top level schematics are explained in greater detail later in this chapter.

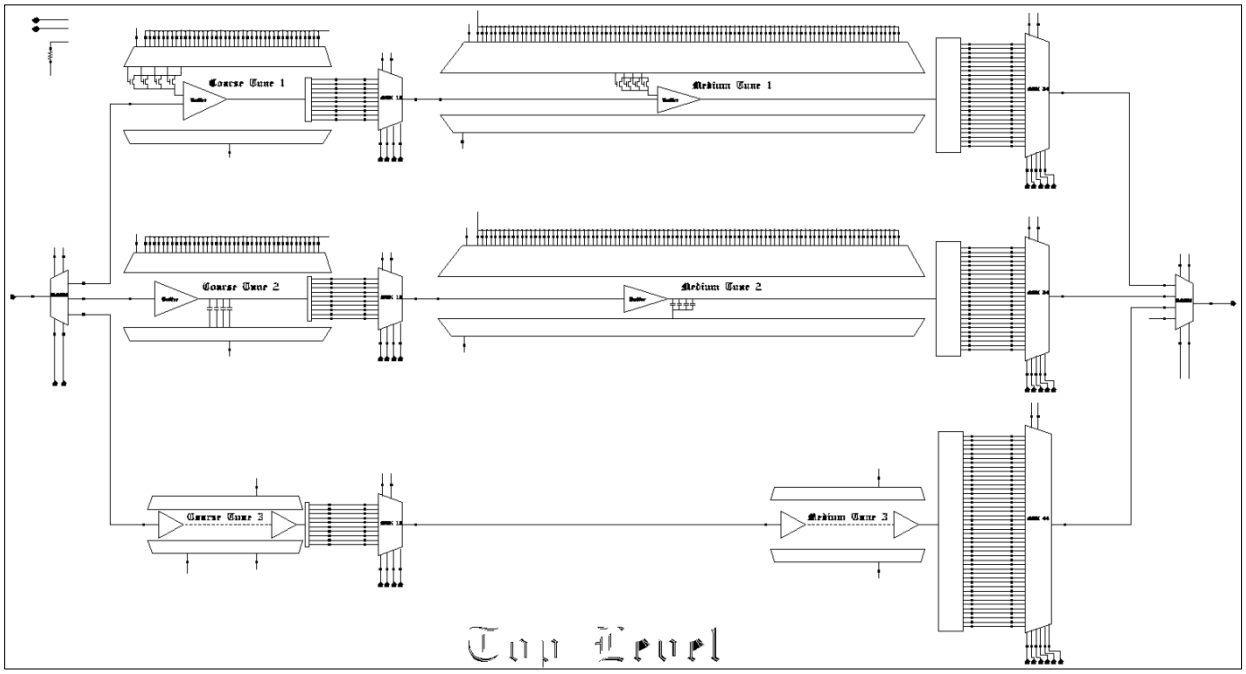

Figure 3.1. Top level schematics of the delay lines consisting of 19 520 MOS transistors

There are 3 delay lines implemented in the schematics. Delay line 1 is current starved, delay line 2 is load biased and delay line 3 is made up of non-tunable buffers (figure 3.1 and figure 3.2). The tunable delay elements were designed with an approximate delay tunability of $\pm$ 10% of the desired intrinsic delay.

The demultiplexer on the input selects which of the delay lines the input signal propagates through. The demultiplexer is implemented to reduce the power consumption so that only *one* delay line starts switching.

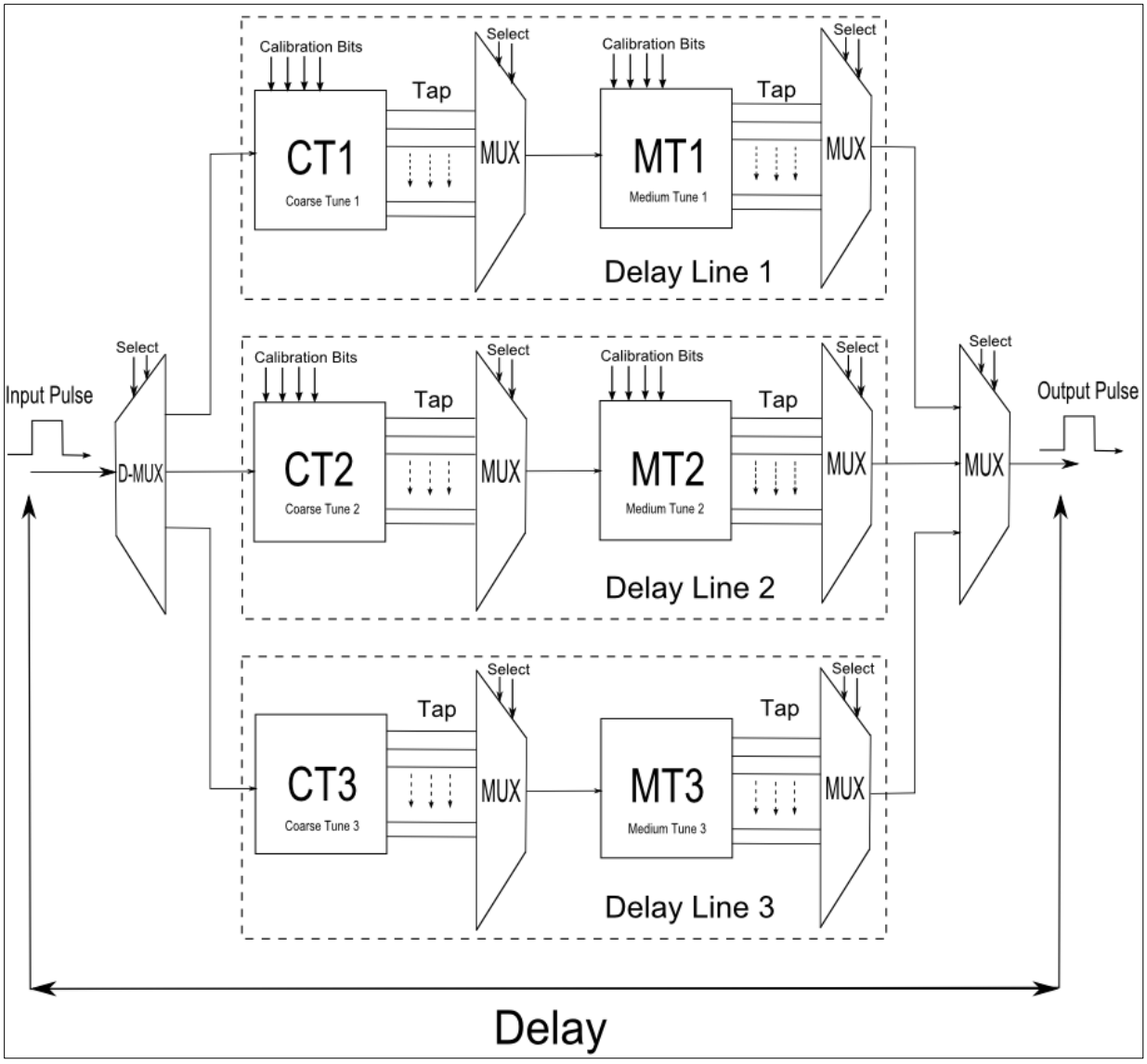

Figure 3.2. Top level block diagram of circuit

Within each delay line there is a coarse tune and medium tune-element as well as two multiplexers to tap the outputs of the elements. The coarse tune element is made up of 10 cascaded coarse tuned delay elements with an intrinsic delay of $1ns$ each, giving it a total delay of $10ns$, matching the specified PRF.

Figure 3.3 displays the three different approaches of creating the coarse and medium tune. Note that $CT1$ and $CT2$ are made up using $1ns$ delay elements whereas $CT3$ is made up using 40 cascaded $25ps$ delay elements to create the $1ns$ delay.

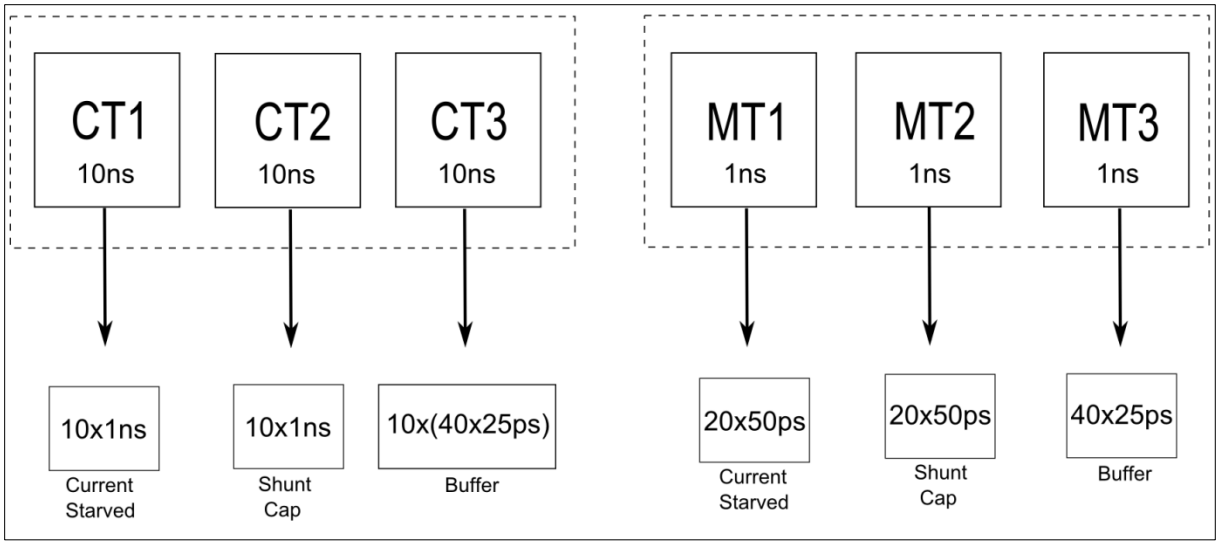

Figure 3.3. Components of coarse and medium tune elements

Each $1ns$ delay element within the coarse tune can be tapped through a multiplexer using the select bits. These select bits therefore determine the total delay of the coarse tune element from $1ns \rightarrow 10ns$ with a step of $1ns$. In addition to the tapping, each of the delay elements making up $CT1$ and $CT2$ has 4 bits of tuning. This means that these coarse tune elements can output via the multiplexer a delay from $1ns \rightarrow 10ns$ with a step of $1ns$ in addition to the 4 bits of tuning that can change the intrinsic delay of the $1ns$ delay elements, compensating for environmental variations and mismatch. Figure 3.4 displays the inside view of the $CT1$ element.

Obviously layout and routing is a main issue for timing, especially for the faster medium tune elements explained later on.

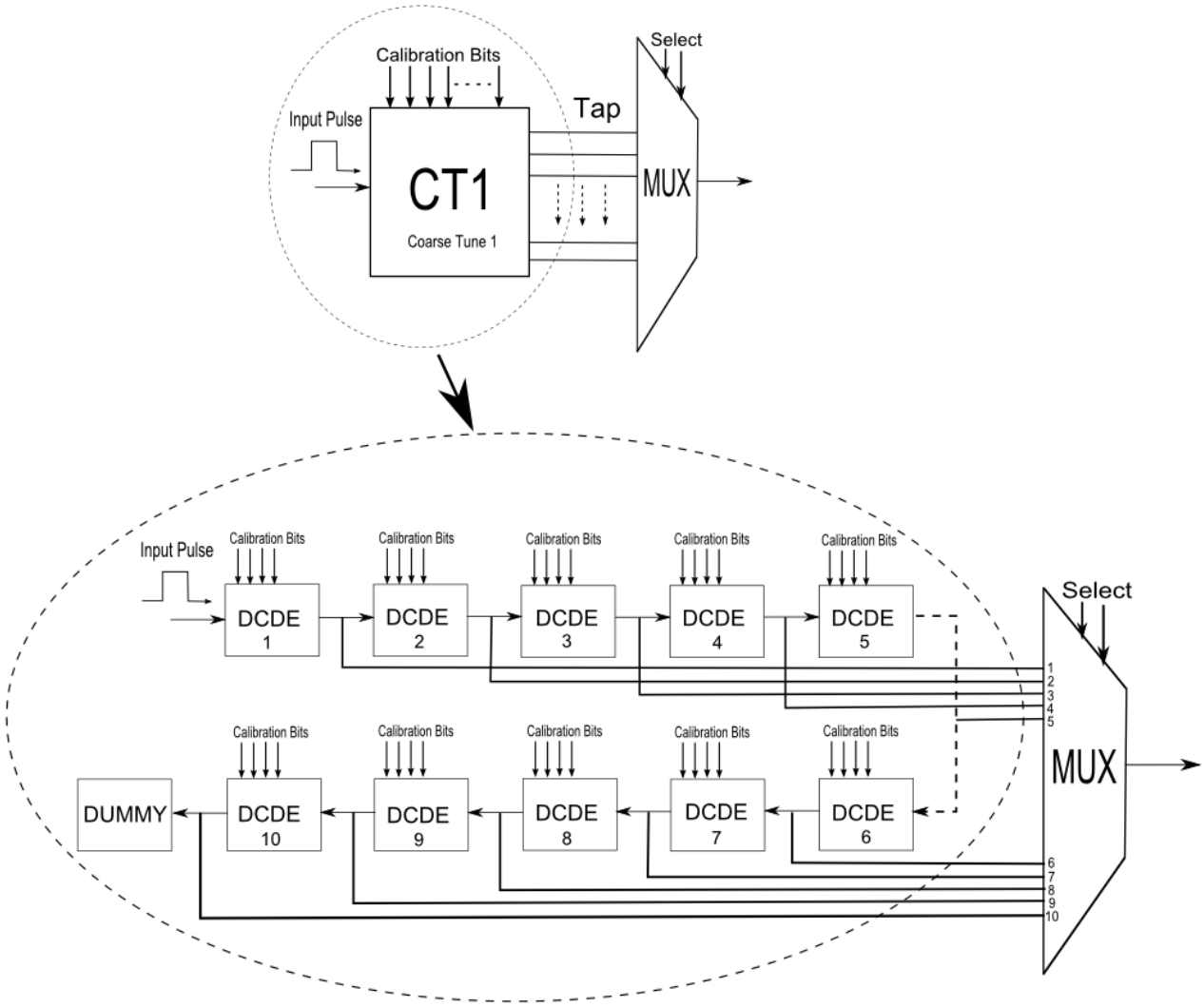

Figure 3.4. The inside of a coarse tune element

A medium tune elements is basically the same as the coarse tune elements but the delay elements inside are much quicker than those in the coarse tune elements with an intrinsic delay of $50ps$. The medium tune therefore has 20 delay elements to make up for *one* coarse tune element (figure 3.3). This means that the total delay line is programmable from $1050ps \rightarrow 11ns$ with a step of $50ps$ in addition to the 4-bits tunability of each delay element, not taking the multiplexer delay into account.

One exception is the non-tunable buffer delay line which has 40 delay elements because they are approximately twice as fast as the $DCDEs$ with an intrinsic delay of $25ps$. This means that the delay line is programmable from $1025ps \rightarrow 11ns$ with a step of $25ps$. It is to be noted that the programmability comes from the higher number of taps due to a higher number of delay elements and larger multiplexers, not the fact that the delay elements themselves are tunable.

After the signal has propagated through the delay line a 3:1 multiplexer selects which of the delay lines that is passed through to the output. This multiplexer should have the same setting as the input demultiplexer.

3.4 Demultiplexer

Figure 3.5 illustrates the schematics of the 1:3 demultiplexer.

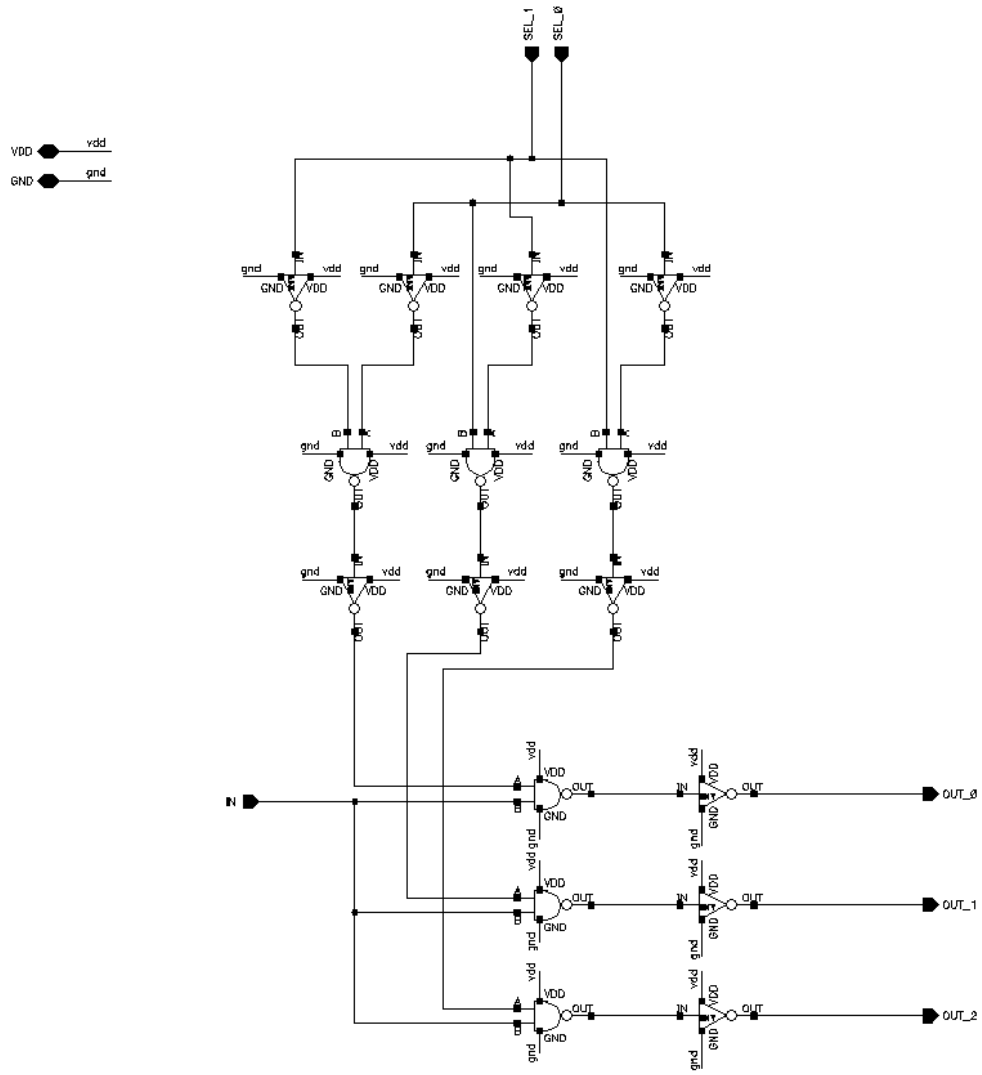

Figure 3.5. Schematics of 1:3 demultiplexer circuit

The 1:3 demultiplexer is used to select which delay line the input pulse is passed through to and is made up using two logic cells, $NAND$-gates and inverters. Inputs "SEL_0" and "SEL_1" activates one of the outputs to be passed through depending on the select bit values, see table 3.1.

The demultiplexer is originally a 1:4 demultiplexer as it has 2 select bits but it has been modified to only have the required 3 outputs to save silicon area. An important design parameter of the demultiplexer is to uniformly distribute the critical signal paths in the layout to create an equal delay for all delay lines. The $NAND$ and inverter logic cells used in the demultiplexer is explained later on.

Select bits "SEL_0" **and** "SEL_1"	**Output activated**	**Comment**
"**00**"	"OUT_0"	$Delay\ line\ 1\ activated$
"**01**"	"OUT_1"	$Delay\ line\ 2\ activated$
"**10**"	"OUT_2"	$Delay\ line\ 3\ activated$
"**11**"	"XX"	$Not\ valid\ select\ value$

Table 3.1. Truth diagram for the 1:3 demultiplexer

3.5 Coarse Tune

The coarse tune elements represent the highest delay on the delay line with an intrinsic delay of $10ns$. Both the current starved and shunt capacitor coarse tune is made up using $10\ x\ 1ns$ delay elements, not including the dummy, to create the main delay of $10ns$, matching the specified PRF. A dummy is inserted after the last delay element to ensure equal load capacitance, thus equal delays, for all the delay elements. The non-tunable buffer coarse tune is also made up using $10\ x\ 1ns$ delay elements, but the difference is that the $1ns$ delay elements are made up by cascading $40\ x\ 25ps$ medium tune delay elements. This means that the non-tunable buffer coarse tune and medium tune are made up of the same delay element, only adjusting the cascading factor.

The $1ns$ delay elements are referred to as the coarse tune delay elements and are explained below. There are 3 different coarse tune delay elements, one for each delay line. Schematics of $CT1, CT2$ and $CT3$ are added in appendix B.

3.5.1 Current Starved Coarse Tune Delay Element

A commonly used current starved delay element is depicted in figure 3.6. A drawback with this architecture is that the signal path is constantly changing with the input vectors meaning that it is non-monotonic, not very linear and it is hard to determine the actual delay for a given input vector [1].

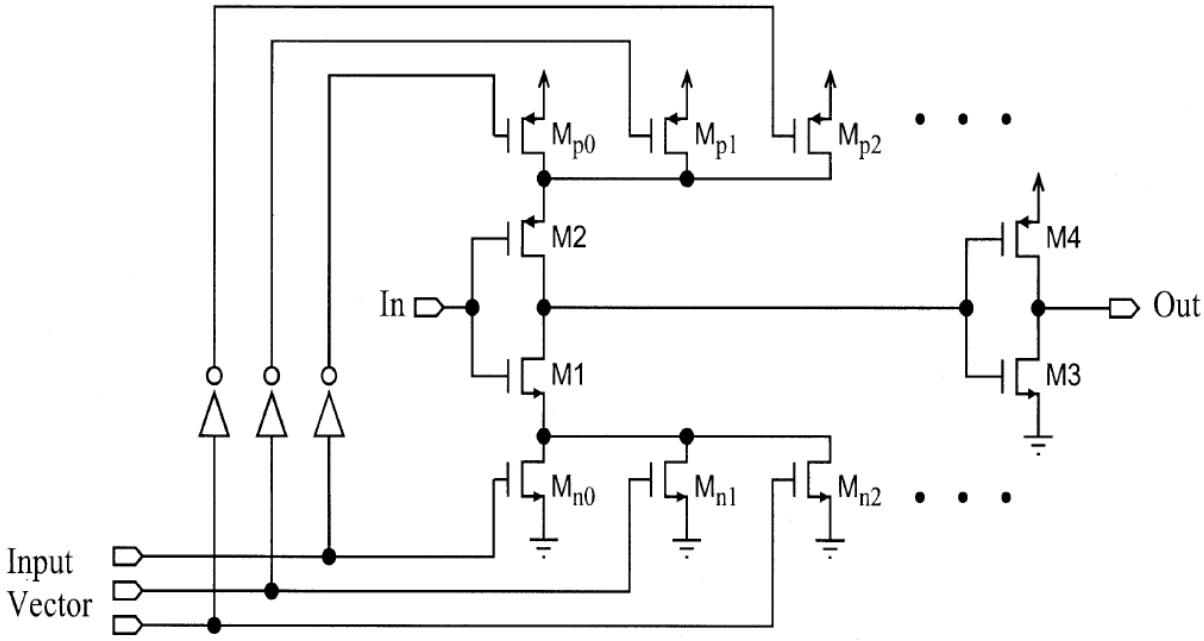

Figure 3.6. A commonly used digitally controllable current starved delay element [1]

An improved current starved architecture and the originally designed architecture is shown in figure 3.7. It has greater linearity, monotonicity and temperature dependence due to the same signal path for all configurations which is less sensitive to process variations and parasites.

In contrast to the architecture shown in figure 3.7 it is to be noted that the new architecture only controls the rising edge of the signal. This gives rise to asymmetric behavior meaning that the pulse passed through the delay element suffers from pulse width reduction.

The main core of the delay element is the current starved buffer, here represented by transistors $M7 -> M11$. The discharging current of the buffer is controlled by a current mirror made up of transistors $M6$ and $M9$, where $M9$ is acting like a current source. The current controlling the current mirror can be adjusted with transistors $M1 \rightarrow M4$. Transistor $M5$ is always on and its W/L ratio determines the maximum delay for the delay element. By turning on more of the adjusting transistors $M1 \rightarrow M4$ more current will flow to the drain of transistor $M6$ as they are connected in parallel. This gives $M6$ and $M9$ a higher gate voltage which in return increases the discharging current thus reducing the delay. With the 4 digital input bits A, B, C and D, 16 different delay configurations can be achieved.

Capacitor $C0$ is a decoupling capacitor with a purpose of reducing the noise from other parts of the power supply lines and to supply the delay elements with sufficient power when they are switching. As the capacitors can respond very quickly to changing current demands they are used as energy storages all over the circuit.

The capacitors are charged up by the power supply when the switching activity is zero and provide sufficient power quickly when the input pulse arrives. This is combined with a grid of VDD and GND supply lines to supply power immediately on demand [58]. This can be seen implemented in the layout in appendix A.

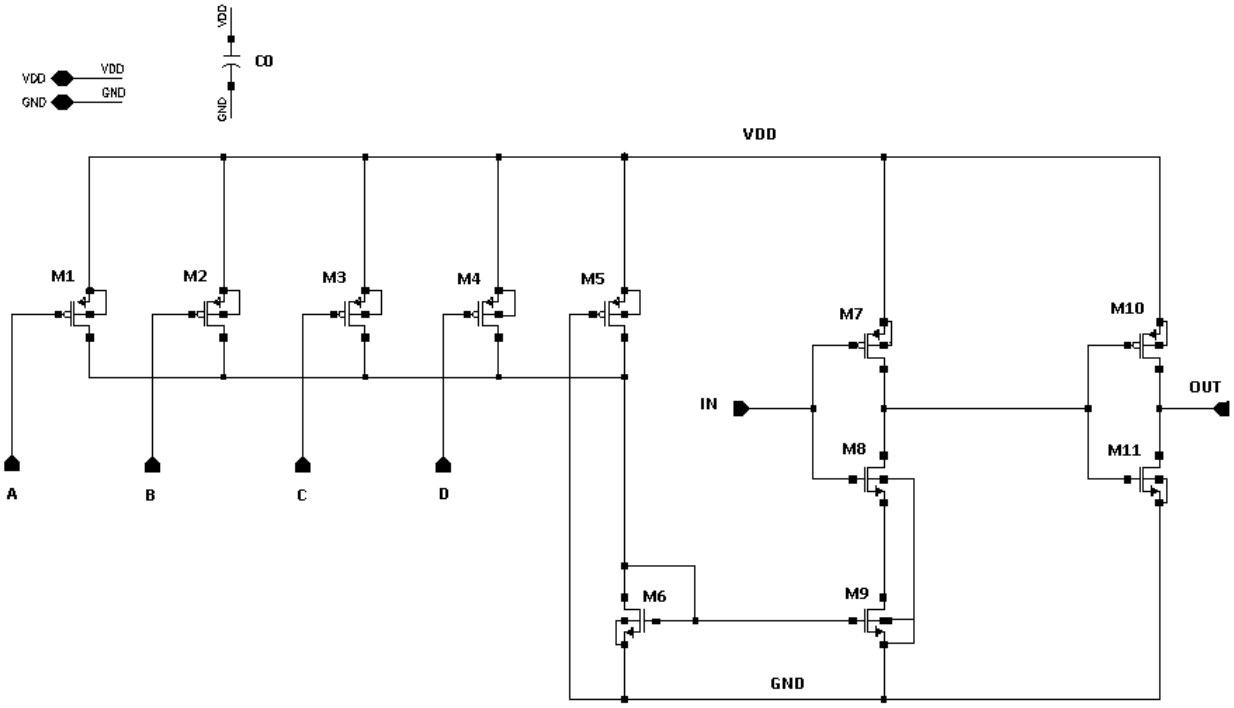

Figure 3.7. Improved current starved delay element architecture

To design and size the current starved delay element some design procedures are useful to simplify the process [59]:

- The transistor sizing of the output inverter ($M10$ and $M11$) is mainly determined by the load capacitance. In this case where the delay elements are cascaded, the transistors $M7$ and $M8$ of the next delay element becomes the load.
- Transistor $M9$ should be much smaller than $M8$ to control the discharging current.
- Transistor $M6$ and $M9$ can be the same size.
- The number of digitally programmable $pMOS$ transistors, in this case 4 ($M1 \rightarrow M4$), can be raised/lowered to obtain more/fewer delay configurations. The number of delay configurations N with this binary weighting of m transistors can be expressed by: $N = 2^m$.

- Transistor $M5$ is always on and is sized to achieve the maximum desired delay.
- When transistor $M5$ is sized for maximum delay, *one* $pMOS$ transistor (call it $M0$) should be added in parallel with $M5$ to simulate all of the adjusting transistors and sized for the minimum desired delay.
- After this $M0$ should be broken into m number of transistors in a binary weighted form:

$$\left(\frac{W}{L}\right)_{M_i} = \frac{2^{i-1}}{2^m-1}\left(\frac{W}{L}\right)_{M0} \tag{3.1}$$

for $i = 1,2,3,\dots,m$

- After this the circuit can be tested/simulated and fine-tuned to give exactly the desired delays, although it is not necessary to become too perfectionistic as the parasitic components will limit the resolution anyways.

As the current starved coarse tune delay element has a quite long intrinsic delay of $1ns$ in a quite fast $90nm$ process, it means that the element has to be heavily starved. Heavy starving gives rise to very slow rising edges in nodes within the buffer which in return increases the jitter. The linearity of the delay element also suffers as the non-linear current source transistor has to operate over a wide range. Therefore a new approach to designing the element was initiated. Instead of starving the buffer so heavily, the idea was that extra buffers could be introduced to increase the intrinsic delay, thus reducing the current starving of the buffer. This would in return result in faster rising edges (higher slew rate) and less jitter. The tradeoff is a slight higher temperature dependency due to more components. The number of extra buffers to insert had to be simulated on to come close to, but not over, the desired maximum delay. In this case the optimum configuration was achieved using 28 extra buffers to make up the element as illustrated in figure 3.8.

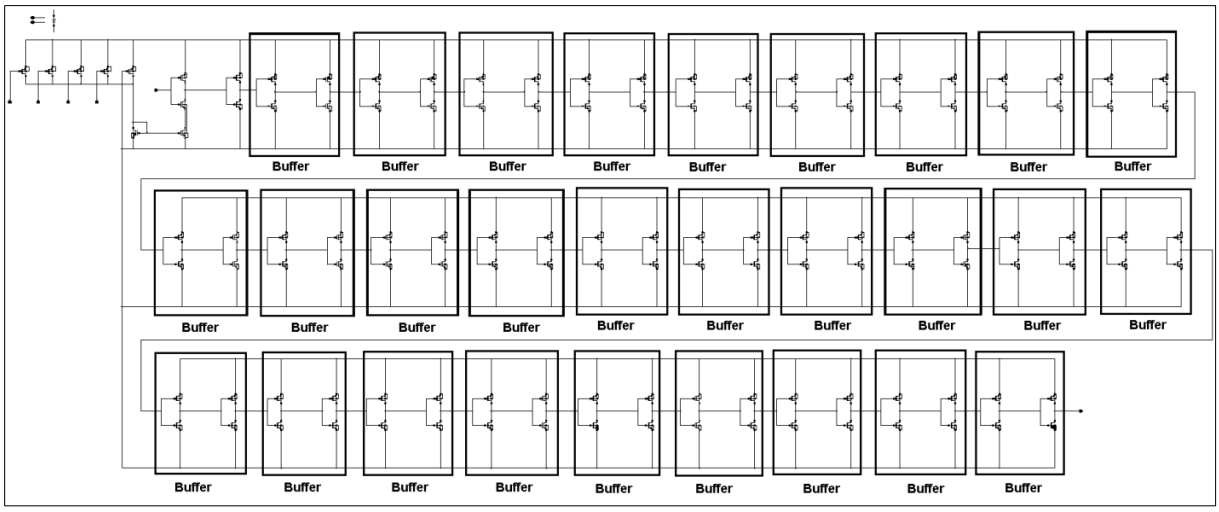

Figure 3.8. Further improved current starved coarse tune delay element architecture

After inserting the extra buffers, the transistors in the main current starved buffer were resized using the same procedures as listed above, to achieve the desired delay. The intrinsic tunability was designed to achieve $\pm 10\%$ of the main intrinsic delay.

3.5.2 Shunt Capacitor Coarse Tune Delay Element

Figure 3.9 depicts the originally designed shunt capacitor circuit. The main element is the buffer consisting of transistors $M1$, $M2$, $M11$ and $M12$. Compared to the shunt capacitor delay element shown in figure 2.29 [50] this design is a bit different. Here the transistors $M7 \rightarrow M10$ are all equal and very small $nMOS$ capacitors to realize a capacitive load while the gate-source capacitance (C_{gs}) of transistors $M3 \rightarrow M6$ determines the actual load biasing, not the W/L ratio. The load is digitally switched in compared to an analog gate voltage determining the charging/discharging current. The main reason for this design procedure was that during testing this was found to reduce the overall size of the transistors as the sizing of the input transistors had a greater effect on the delay. Transistors $M3 \rightarrow M6$ all have the same W/L ratio but their area is binary weighted from the digital inputs $A \rightarrow D$. Therefore the delay will change with the input vector as extra load capacitance is switched in. The $nMOS$ gate-source capacitance can be expressed by [60]:

$$C_{gs} \cong \frac{2}{3} W L C_{ox} \tag{3.2}$$

where W is the transistor width, L is the transistor length and C_{ox} is the gate-oxide capacitance.

Capacitor $C2$ is a decoupling capacitor and is combined with a grid of VDD and GND supply lines to supply power immediately on demand.

Similar to the coarse tune current starved architecture, the coarse tune shunt capacitor could also be improved by inserting buffers to improve the rise times. This would reduce the capacitive load, thus reducing some of the silicon area for the capacitors, even though the total area would increase by the added buffers. It would also improve the slew rate which reduces jitter. A total of 32 extra buffers were inserted into the improved architecture (figure 3.10). After inserting the extra 32 buffers the input transistors $M3 \rightarrow M6$ were resized to create the desired delay. Since this architecture affects both the falling and rising edge of the input pulse it does not have any issues with pulse width reduction compared to the current starved architecture. As a matter of fact the pulse width actually increases slightly as the second inverter of the buffer was made a bit smaller to reduce the overall delay.

The intrinsic tunability was, as for the current starved architecture, designed to achieve $\pm$ 10% of the main intrinsic delay.

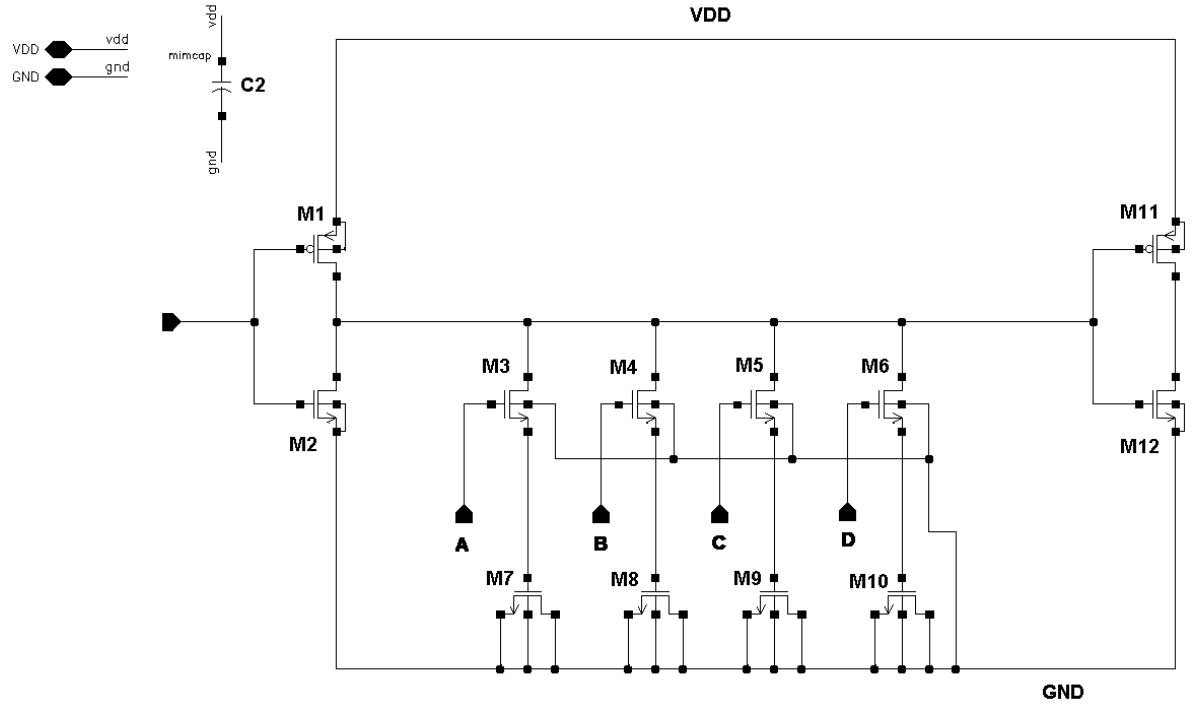

Figure 3.9. Originally designed shunt capacitor coarse tune delay element

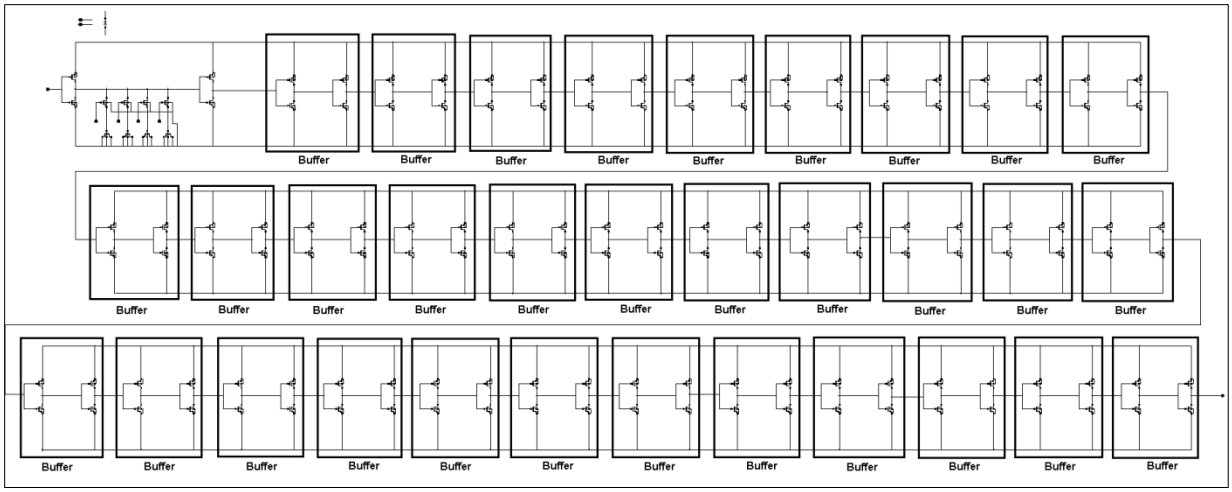

Figure 3.10. Improved shunt capacitor coarse tune delay element architecture

3.5.3 Buffer Coarse Tune Delay Element

The buffer coarse tune delay element is the simplest of the three architectures. It only consists of buffers with no intrinsic tuning and the buffers are designed as fast as possible to increase the slew rate and reduce jitter. Note that the buffers are the same buffers used in the medium tune buffer delay element. It has an intrinsic delay of $25ps$ and to make up the $1ns$ desired delay, $40x25ps$ buffers are cascaded as illustrated in figure 3.11.

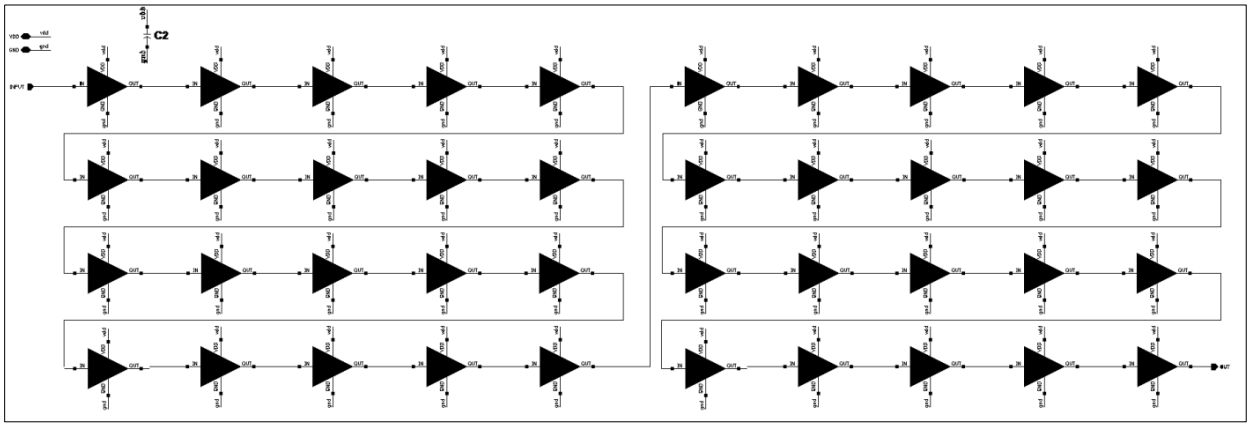

Figure 3.11. Buffer coarse tune delay element schematics

Since the buffer coarse tune delay element has no intrinsic tuning it is very simple and easy to implement. The buffer coarse tune delay element is only a scaled up version of the buffer medium tune delay element, whereas the current starved and shunt capacitor delay elements has two different designs for medium and coarse tune.

3.6 Medium Tune

The medium tune delay elements are much quicker than the coarse tune delay elements and therefore supplies higher level of tunability and resolution to the delay lines. Both the current starved and shunt capacitor medium tune is made up using $20\ x\ 50ps$ delay elements, not including the dummy, to create a minimum of *one* coarse tune element ($1ns$). The non-tunable buffer medium tune delay element is approximately twice as fast as the others, meaning it is made up of $40\ x\ 25ps$ delay elements. It could be designed to have the same delay as the others by resizing it to be slower, but all the medium tune delay elements are designed to be as quick as possible as this reduces rise times and jitter.

It also makes it able to have twice as many taps in the delay line which translates to double the delay configurations.

To ensure that all the cascaded medium tune delay elements makes up for at least one coarse tune delay element for all alters some extra delay elements are added as insurance. Schematics of $MT1, MT2$ and $MT3$ are added in appendix B.

3.6.1 Current Starved Medium Tune Delay Element

In the current starved medium tune delay element the design and functionality is the same as the coarse tune delay element without the extra added buffers. It is resized to have a higher slew rate than the coarse tune element and the shortest delay possible in this process; therefore it is no need to insert extra buffers. Resizing the delay element to achieve a lower intrinsic minimum of $50ps$ means that the transistor sizes are different than the coarse tune but other than that the design is equal to figure 3.7.

3.6.2 Shunt Capacitor Medium Tune Delay Element

The shunt capacitor medium tune delay element has the same architecture and functionality as the shunt capacitor coarse tune delay element. The medium tune delay element has no extra buffers and is resized to create the $50ps$ intrinsic delay, other than that is equal to the coarse tune architecture displayed in figure 3.9.

3.6.3 Buffer Medium Tune Delay Element

The buffer medium tune delay element is the main building block used to make up both the medium tune and coarse tune delay elements as discussed previously. The buffer is designed to be as fast as possible with a desired intrinsic delay of $25ps$ and the ratio between the $pMOS$ and $nMOS$ transistor sizing is adjusted to make the device more symmetric. This makes sure that the buffer does not suffer from any pulse width reduction. Figure 3.12 depicts the schematics of the buffer.

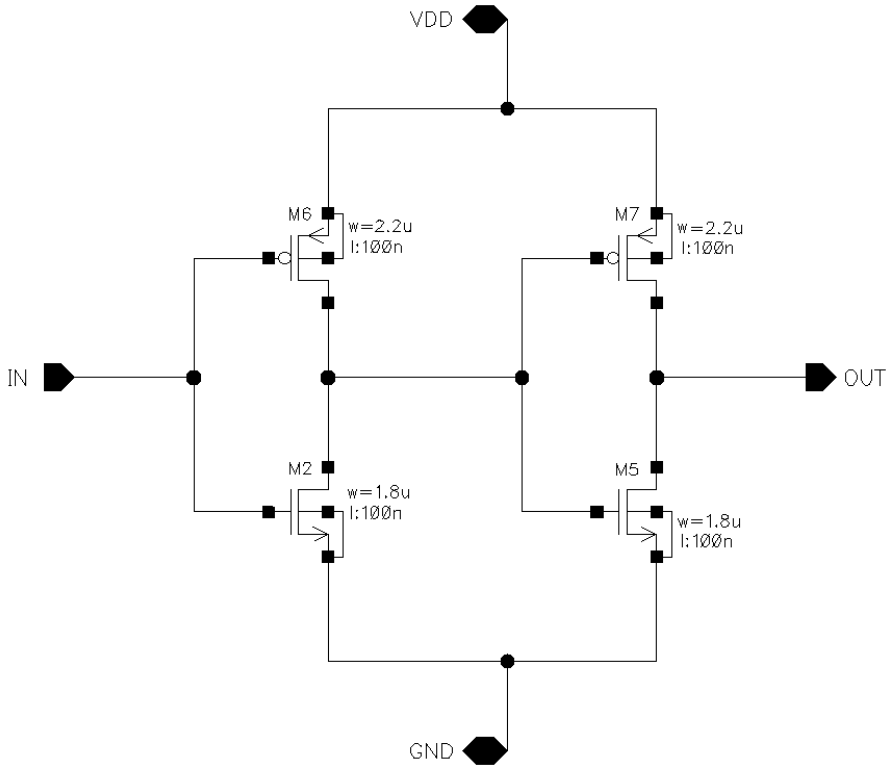

Figure 3.12. Schematics of buffer medium tune delay element

3.7 Multiplexer

The multiplexer is an important part of the design. This determines the delay of the delay lines by tapping the outputs of the coarse and medium tune element. It is also one of the most complicated designs due to large multiplexers with critical signal path considerations in layout. Figure 3.13 shows the 2:1 multiplexer which is used as a building block to create the larger multiplexers.

The 2:1 multiplexer is made up by using 3 logic cells; the inverter, $NAND$-gate and NOR-gate. A difficult but crucial design element is to achieve equal signal paths (delay) for all configurations. From figure 3.13 this relates to equal signal path for both input signals "IN_0" and "IN_1" to the output. Select signal "SEL_0" determines which of the $NAND$-gates that is activated and therefore which of the input signals that are propagated through to the output.

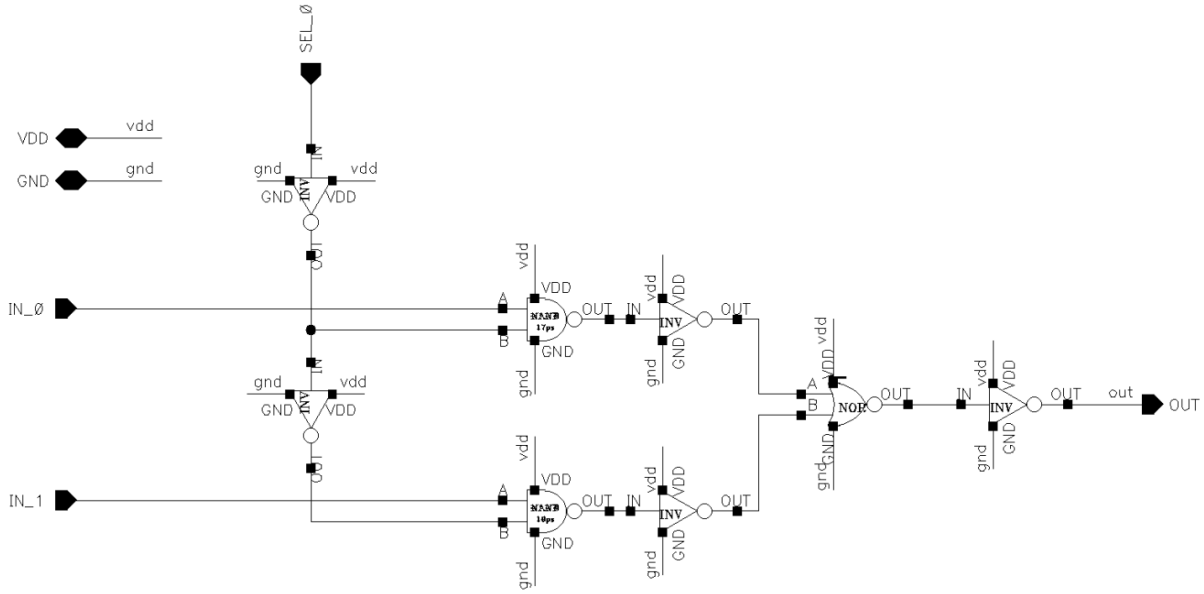

Figure 3.13. 2:1 multiplexer building block

To create larger multiplexers it is possible to combine several smaller multiplexers. Figure 3.14 depicts how a 4:1 multiplexer is made using $3\ x\ 2$:1 multiplexers. This can be extended as far as needed as long as you use binary valued multiplexers to create them ($2,4,8,16,32$ etc.). It is possible to create non-binary value multiplexers as seen in table 3.2, where none of the multiplexers used are binary valued. This is to save silicon area. For instance $CT1$ needs a 10:1 multiplexer to tap its 10 outputs. The closest multiplexer is therefore a 16:1 multiplexer but it is a waste of area to implement a 6 input larger multiplexer than required. It is to be noted that the created non-binary valued multiplexers cannot be used to create larger multiplexers. Therefore the non-binary valued multiplexers have to be made up of binary valued multiplexers. As an example a 20:1 multiplexer cannot be created using $2\ x\ 10$ multiplexers. It *can* be created but it will not have an intuitive monotonic truth table. When creating a non-binary valued multiplexer some delay elements has to be added to the smaller multiplexers as the smaller cells has a shorter delay (figure 3.15). The tradeoff to using non-binary valued multiplexers is increased uncertainty due to unmatched cells and different routing paths.

Since the multiplexers also have intrinsic propagation delay, which is quite significant as the multiplexers are quite large, they will add unwanted delay to the delay lines. If the delay lines are used in a radar system to determine the delay from TX to RX this could be compensated for by adding the same fixed delay to TX.

Table 3.2 indicates the different multiplexer sizes used in the coarse and medium tune elements in the delay lines. Note that the $MT3$ multiplexer size stands out from the rest as this has delay elements with shorter intrinsic delay than the digitally controllable delay elements.

Element	Multiplexer Size
CT1	10
CT2	10
CT3	10
MT1	24
MT2	24
MT3	44

Table 3.2. List of multiplexer sizes used in the delay lines

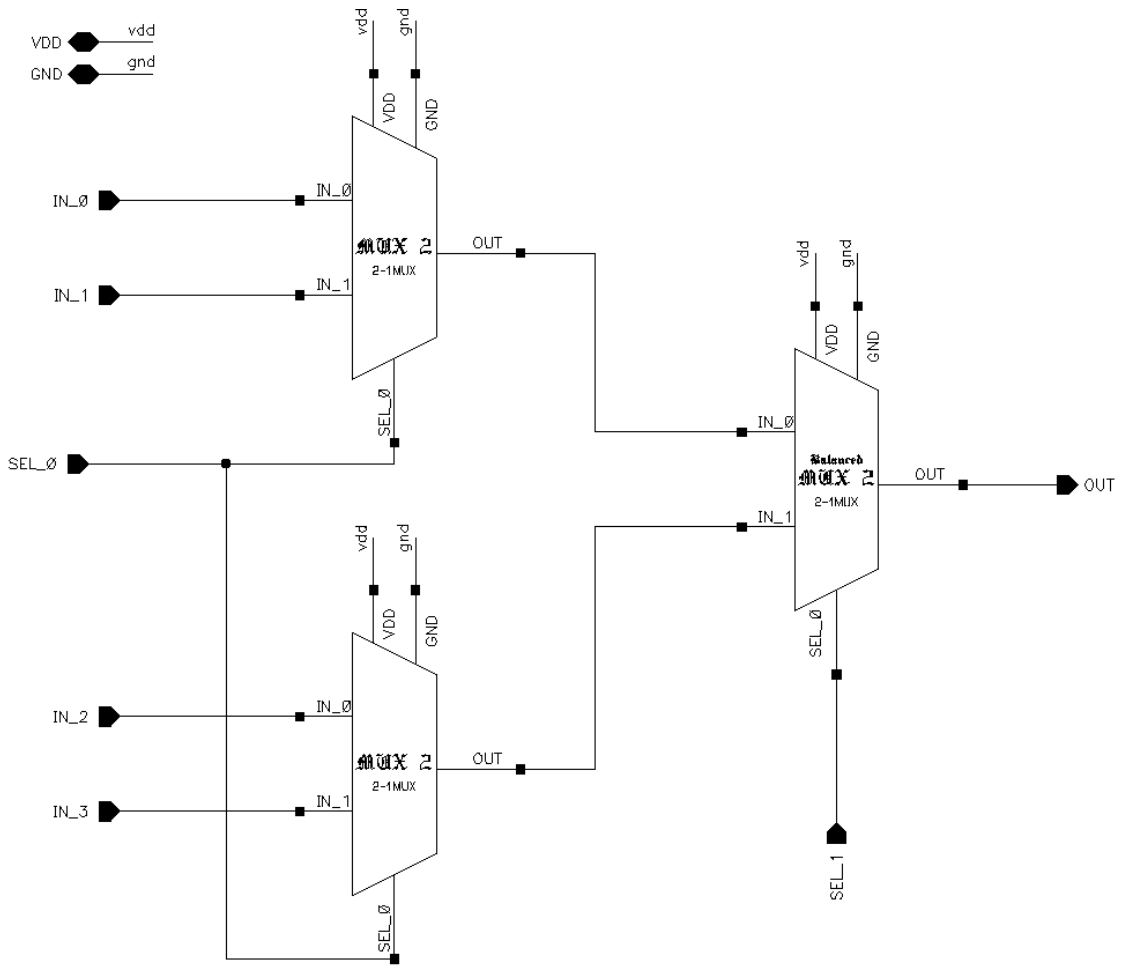

Figure 3.14. Realizing a 4:1 multiplexer using 3 x *2:1 multiplexers*

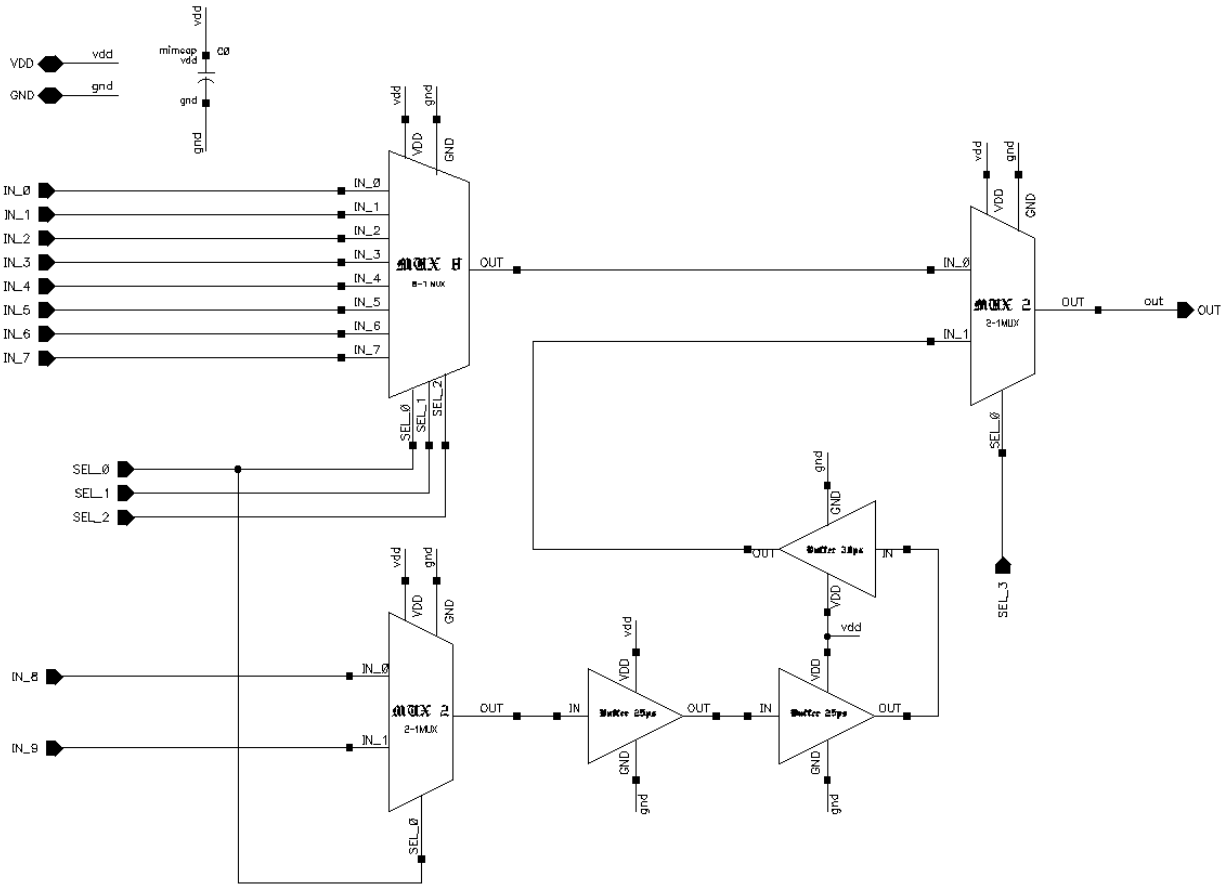

Figure 3.15. Non-binary valued 10:1 multiplexer

3.8 Inverter

The inverter cell is used in the digital logic in both the multiplexer and demultiplexer making it a part of the signal path. Figure 3.16 displays the schematics of the inverter. Transistor sizing is kept small, but above minimum, to save silicon area and still keep capacitances low while reducing some short-channel effects. The inverter used in the digital logic has not been designed to be symmetric which could be a disadvantage related to the pulse width reduction mentioned previously. On the other hand, considering that the device is very fast, this should not be a big issue. To keep the inverter approximately symmetric, referring to equal charging and discharging-current due to different mobility, different ratios of $pMOS$ and $nMOS$ widths should be simulated on while measuring the rise and fall-time of the device.

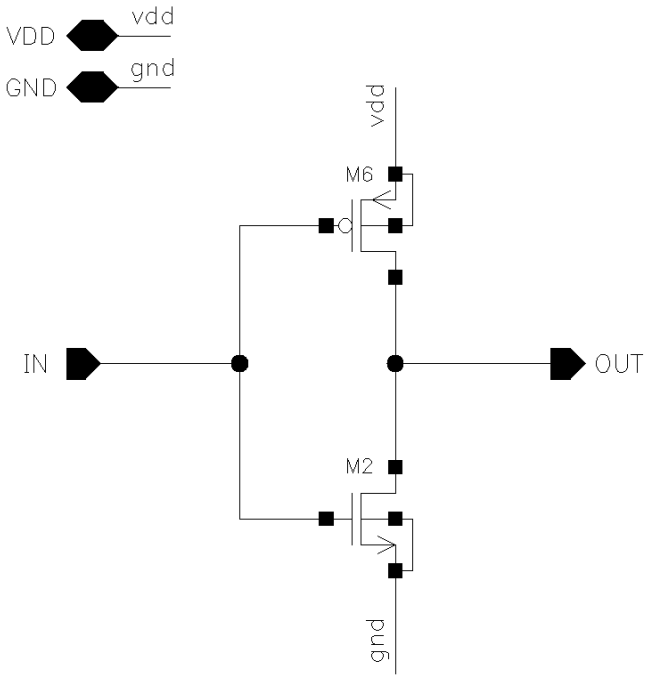

Figure 3.16. Inverter used in the digital logic

3.9 NAND

The $NAND$ gate used in the multiplexers and demultiplexers is illustrated in figure 3.17. Instead of the conventional 4 transistor architecture shown in figure 3.18 a balanced 6 transistor $NAND$ gate was used. The idea was that the two $nMOS$ transistors could be split into two parallel branches with halved transistor widths, where one of the branches was inverted. This would create a more balanced delay between the logical transitions since the transistors are stacked.

The functionality is equal to an ordinary $NAND$ gate, where both input A and B has to receive a logical "1" to produce a logical "0" on the output. To achieve the required AND gate function for use in the multiplexers and demultiplexers, since the $CMOS$ logic cells are complementary, the $NAND$ gate's output is connected to an inverter.

Symmetry between charging and discharging currents is not considered using equal $pMOS$ and $nMOS$ widths. This might result in some pulse width reduction but it should not be a big issue considering the device is fast compared to the delay elements in the delay line.

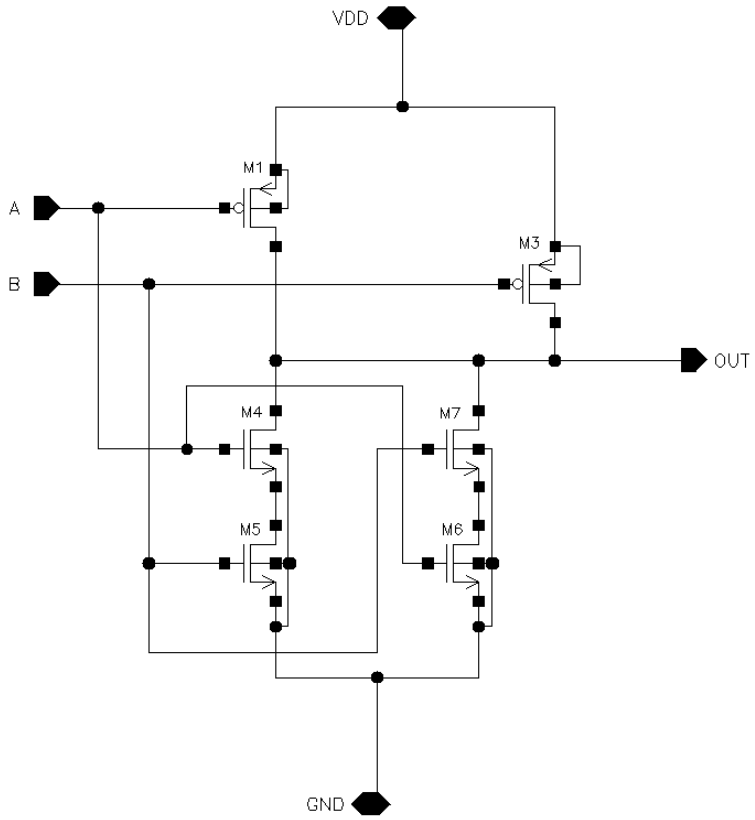

Figure 3.17. Balanced 2 input NAND gate

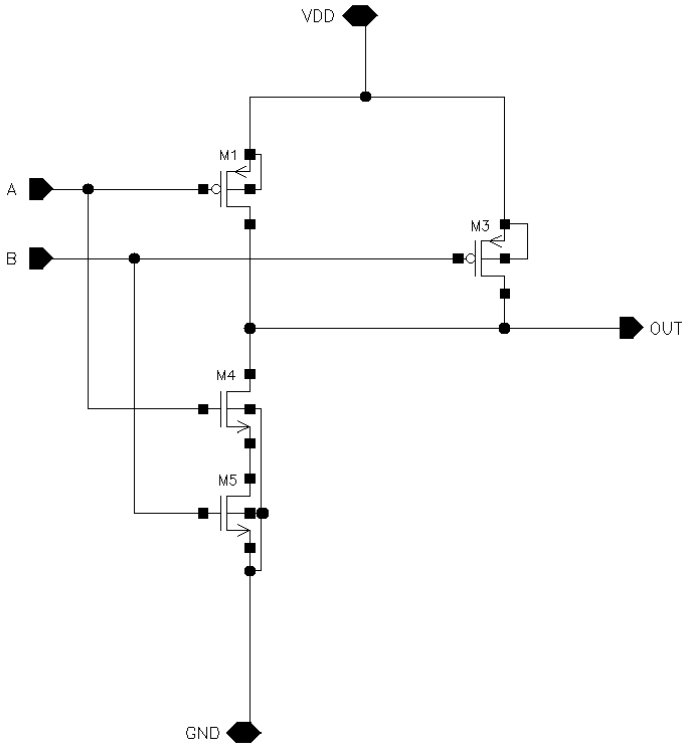

Figure 3.18. Original 2 input NAND gate

3.10 NOR

The NOR gate used in the design is a plain NOR gate with small transistor sizes, as seen in figure 3.19, to maintain fast switching speed and low capacitances. To achieve a logical "0" on the output one or both of the inputs A and B has to be a logical "1". The output is then connected to an inverter to realize the desired OR function required in the digital logic within the multiplexers and demultiplexers.

As for the inverter and $NAND$ gate, the NOR gate has not been designed with focus on device symmetry, which could potentially be a disadvantage as explained previously.

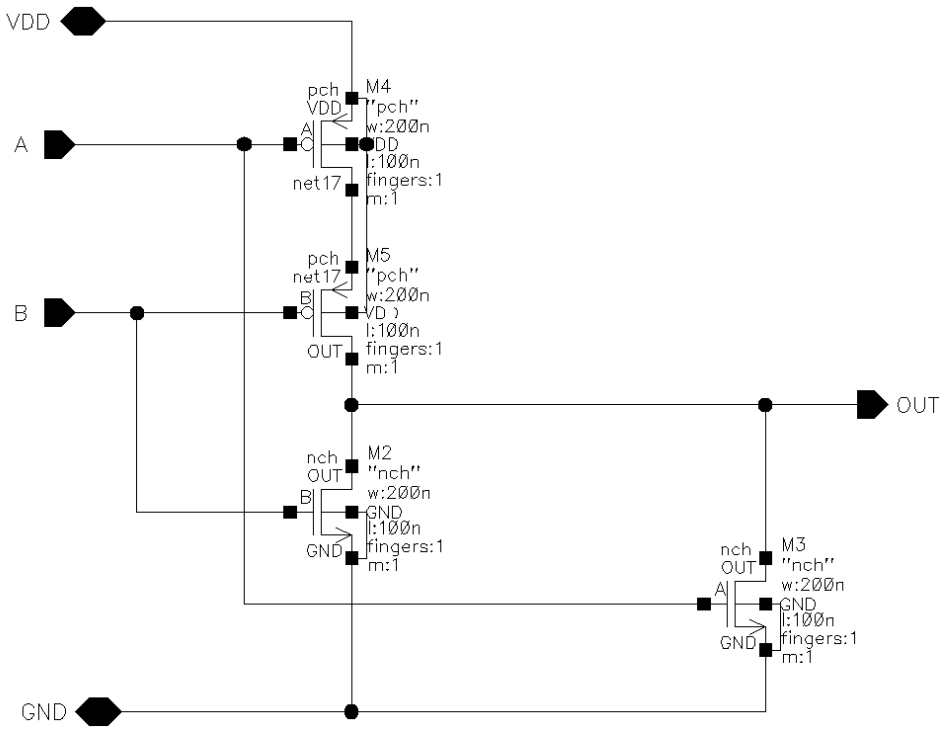

Figure 3.19. 2 input NOR gate

3.11 Reduced Temperature Dependency Delay Element

As static variations such as process variations and mismatch are more easily compensated for as they do not change with time, environmental variations such as temperature variances becomes one of the main concerns for achieving high accuracy, thus techniques for reducing them are of interest.

As previously explained temperature affects the delay elements, making them slower with more leakage current at higher temperatures due to increased channel resistance and thermally excited substrate charge carriers respectively. The proposed solution to reduce the temperature dependency is to use back gate tuning in order to make the transistors faster at higher temperatures, thus eliminating or at least compensating for some of the temperature dependency.

3.11.1 Back gate tuning

Back gate tuning or body biasing can be used to alter the threshold voltage of a MOS transistor and therefore change the intrinsic delay. The threshold voltage can be expressed by [61]:

$$V_{th} = V_{th0} + \gamma\left(\sqrt{V_{SB} + |2\emptyset_F|} - \sqrt{|2\emptyset_F|}\right) \tag{3.3}$$

where V_{th0} is the threshold voltage with $V_{SB} = 0V$, $\emptyset_F$ is the Fermi potential of the body and γ is the body-effect constant:

$$\emptyset_F = \frac{kT}{q}\ln\left(\frac{N_A}{n_i}\right) \tag{3.4}$$

$$\gamma = \frac{\sqrt{2qN_AK_S\varepsilon_0}}{C_{OX}} \tag{3.5}$$

The back gate tuning has many applications and is for instance used in $VTCMOS$ [62]. In $VTCMOS$ the main reason for implementing the back gate tuning is to be able to vary the threshold voltage and therefore reduce leakage currents as well as adjusting the propagation delay, thus adjusting the power consumption.

Figure 3.20 illustrate how the propagation delay of an inverter can be altered using $nMOS$ back gate tuning and figure 3.21 illustrates the schematics set up for $nMOS$ back gate tuning by connecting a voltage to the $nMOS$ bulk.

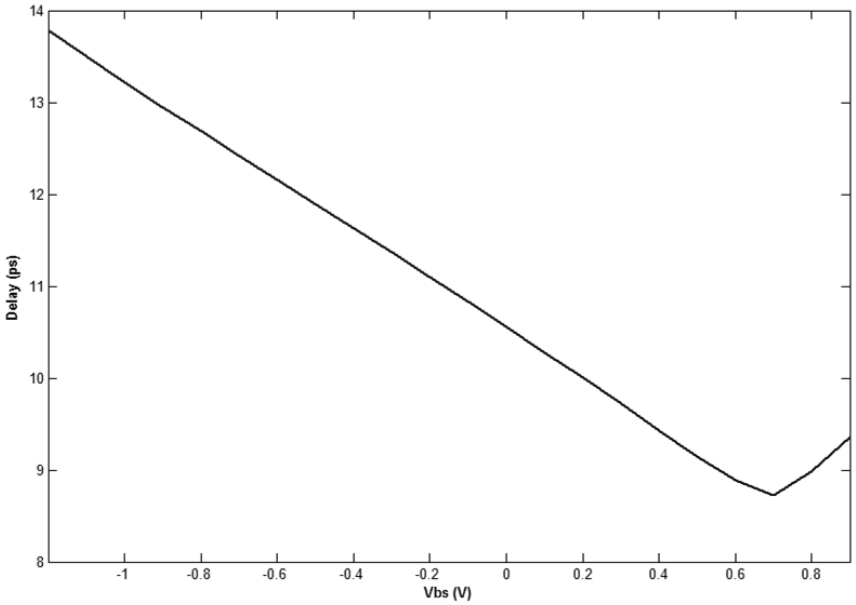

Figure 3.20. Delay as a function of bulk-source voltage for an inverter with $nMOS$ back gate tuning. $pMOS$ (W = 2 µ, L = 100n) and $nMOS$ (W=1µ, L =100n)

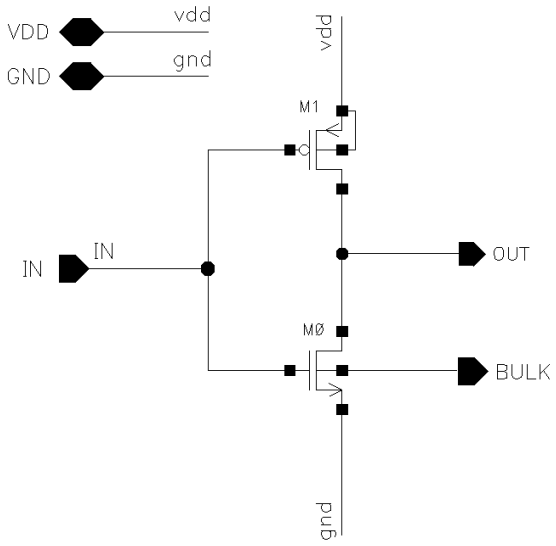

Figure 3.21. Inverter with $nMOS$ back gate tuning

To implement the $nMOS$ back gate tuning to compensate for temperature dependency, it is seen from figure 3.20 that a higher bulk-source voltage $V_{BS}$is needed at higher temperatures to reduce the delay and the temperature dependency. To accomplish this, a voltage dividing circuit with two elements with different temperature dependency connected to the bulk can be considered. Combining these two elements we can constructively utilize the temperature dependency of these two elements to reduce the temperature dependency of the inverter.

As a bonus the inverter should also become quicker as the intrinsic delay decreases when applying a positive V_{BS} to an $nMOS$ transistor. Two different devices with different temperature dependencies are required for this to work. A transistor and a resistor can be used for this purpose as they have different temperature characteristics. Figure 3.22 illustrates how such a voltage dividing circuit, creating a positive V_{BS} as a function of increased temperature, can be realized.

Since the resistor $M1$ has a higher temperature dependency factor than the $pMOS$ transistor $M3$, the voltage drop over $R1$ creates a positive V_{BS} with increasing temperature, which is exactly what is desired. The effect of this bias circuitry is limited to the difference in temperature dependency between the two elements, meaning it is hard to achieve the high voltage drop needed to achieve high compensation for the temperature variations.

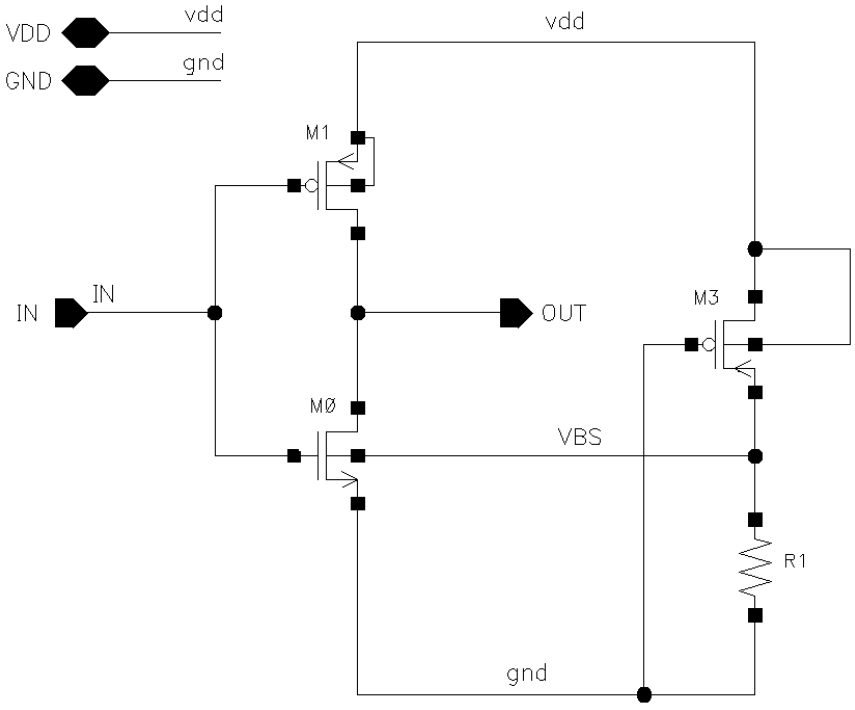

Figure 3.22. Inverter with $nMOS$ back gate tuning circuit

Delay elements are usually created with a buffer as a main core. This means that the pulse of interest, the rising edge, has to go through both an $nMOS$ and a $pMOS$ transistor in the two inverters making up the buffer. Therefore, to keep the device symmetric and even more temperature independent the back gate tuning should be implemented for both the $nMOS$ and $pMOS$. More specifically it should be implemented for the $nMOS$ in the first inverter and the $pMOS$ in the second inverter.

Since the $pMOS$ device has an opposite behavior of the $nMOS$ device, an increasing *negative* V_{BS} with increasing temperature will make the device faster and more temperature independent. Therefore the inversed voltage divider compared to the one implemented for the $nMOS$ device can be used to bias the $pMOS$ bulk.
The $pMOS$ back gate tuning circuit is shown in figure 3.23 and the principle is the same as for the explained $nMOS$ back gate tuning circuit, only inverted and applied to the $pMOS$ bulk.

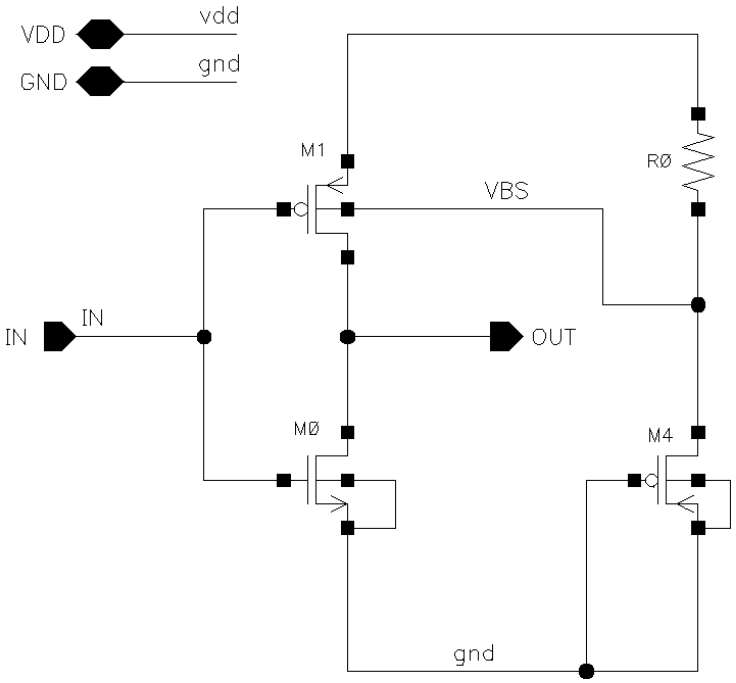

Figure 3.23. Inverter with $pMOS$ back gate tuning circuit

One of the main issues to make this work is to choose a resistor with the right temperature dependency relative to the transistor. The used $tsmcN90rf$ library provides several resistors to choose from. After simulating on the different available resistors the best results were achieved using the "$rnwod$" resistor which is an n-well resistor.

4 Layout

After schematic design has been completed to a level where the performance is satisfactory the next step is to proceed to the layout. Layout is a critical part of the design as parasites are added and the actual “real-world” circuit performance depends on clever solutions as even more non-ideal effects are modeled. It is to be noted that the schematics also includes many of the non-ideal effects.

The linearity and delay of each digitally controllable delay setting could be fine-tuned in the schematics but would probably be pointless as the layout and extracted parasites will alter the fine-tuned delays. Therefore several iterations between schematics and layout are often performed to achieve a satisfactory post layout system performance.

Layout design can be demanding and some techniques and guidelines to follow come in handy. In this chapter some of the most common layout techniques, many of which are implemented, are discussed.

4.1 Layout techniques

Since the system performance is highly dependent on the layout and the intrinsic delay of the $DCDEs$ are important, some layout techniques should be utilized to maintain as much of the schematic performance as possible.

Viewing how the MOS transistor is fabricated makes it easier to understand the parasites existing within the devices and the techniques used to reduce their effect. Figure 4.1 depicts a cross section of a MOS transistor in a simplified version of each step of the fabrication process.

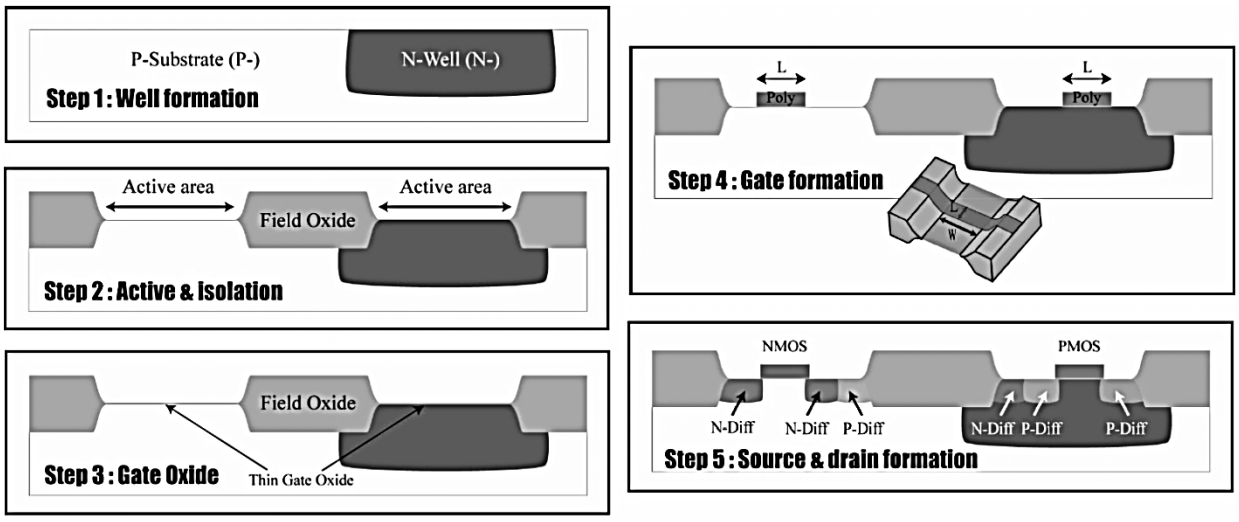

Figure 4.1. Fabrication process steps of a MOS transistor [63]

For instance, to avoid channeling of implanted ions, the wafer is tilted approximately 7° resulting in source/drain asymmetry (figure 4.2) affecting device symmetry.

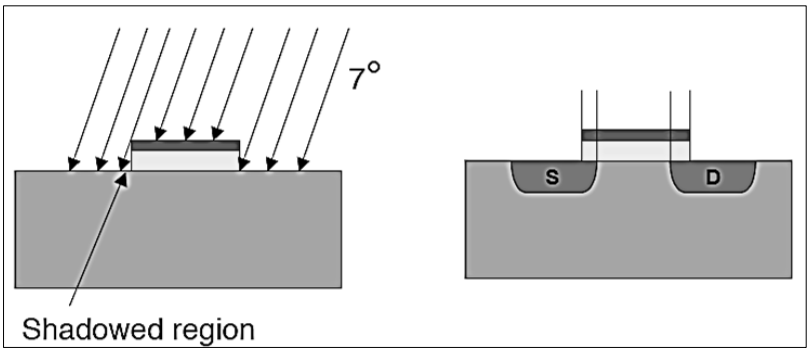

Figure 4.2. Drain and source asymmetry due to fabrication [64]

After the source and drain formation, source and drain connections as well as biasing of the substrate is needed. Ensuring good connections will enhance device performance and improve device asymmetric variations. Using one or few source/drain contacts will increase the impedance and lead to uneven current distribution (figure 4.3) as well as voltage drops. The use of multiple contacts, preferably inserting the maximum number of contacts possible, reduces the resistance and maximizes current flow of the device. The contacts should be placed as close to the source/drain diffusion as possible.

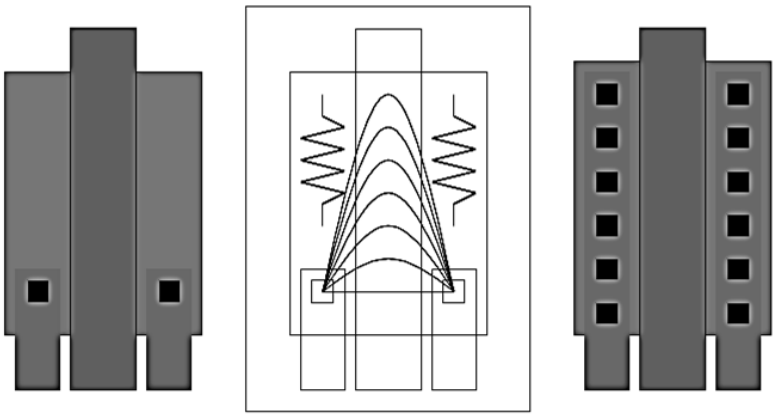

Figure 4.3. Source and drain connections [64]

Figure 4.4 displays a small section of the layout of the $25ps$ buffer displaying how the use of multiple contacts was implemented in the design. The highest number of contacts possible, without resulting in a bad aspect ratio, were inserted as close to the source/drain diffusion as possible.

In the used complementary technology this minimum distance is approximately $400nm$ for n-well contacts and $300nm$ for p-substrate contacts.

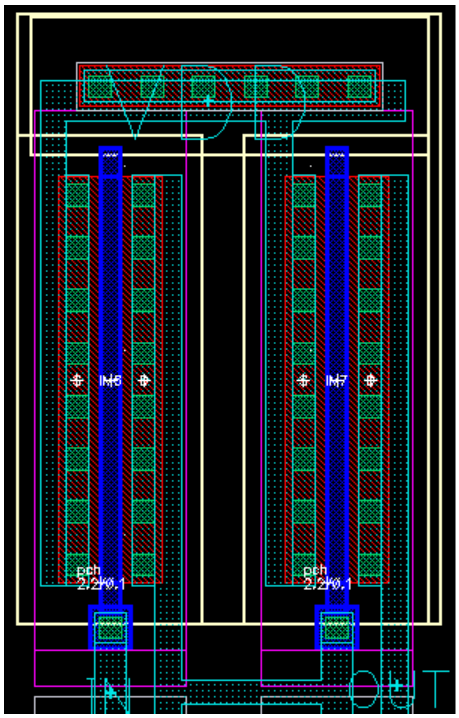

Figure 4.4. Section of the implemented buffer medium tune delay element layout displaying the use of multiple source and drain contacts as well as multiple well contacts

The voltage drop caused by using few source/drain contacts may become an issue as parasitic bipolar transistor exists in a $CMOS$ inverter as illustrated in figure 4.5. These parasitic transistors are connected in a positive feedback loop and can induce latch-up if the transistors are turned on. Increasing the impedance in the substrate (or well) as a result of poor connections may therefore produce the required voltage drop to turn on the transistors and induce latch-up, especially if the current is high as it is in the fastest delay elements.

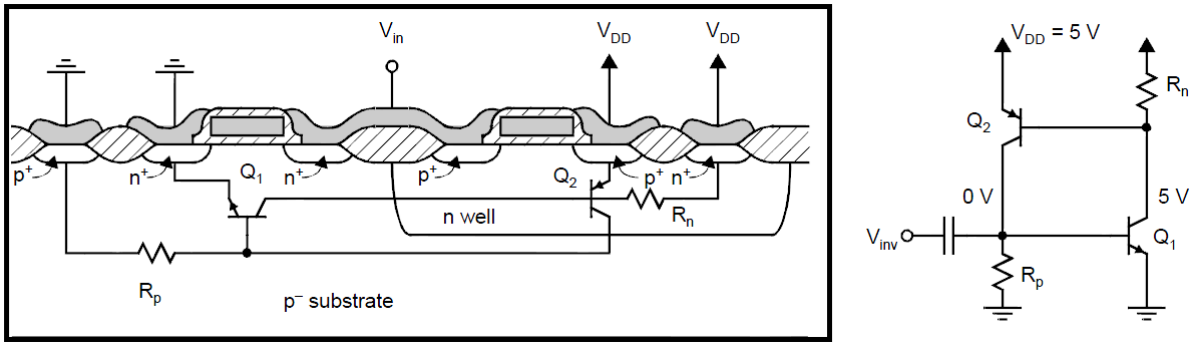

Figure 4.5. Cross section of a CMOS inverter with the parasitic bipolar transistors responsible for the latch-up mechanism and the equivalent circuit [65]

Traditional analog transistors usually have a large W/L ratio giving them a bad aspect ratio. This bad aspect ratio makes a compact layout harder to accomplish. It also has high parasitic diffusion-substrate capacitances and poly gate resistance (C_{SB}, C_{DB} and R_G) making the device slower. Figure 4.6 depicts the parasitic capacitances of a MOS transistor and figure 4.7 shows the parasitic gate impedance for a bad aspect ratio respectively. Viewing these two figures the bad aspect ratio obviously contributes to larger parasitic impedances which in return results in a slower device.

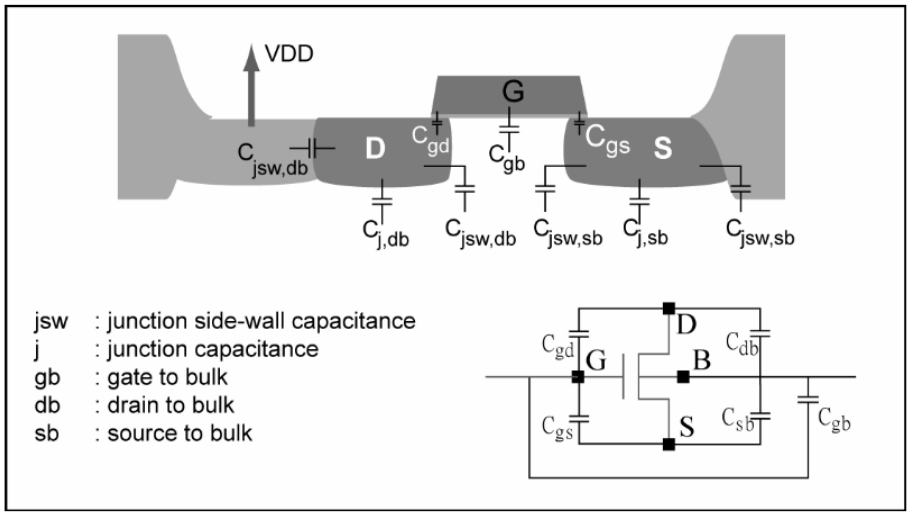

Figure 4.6. Cross section of a MOS transistor with its parasitic capacitances [63]

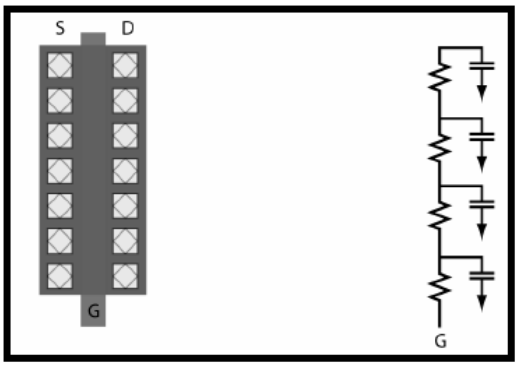

Figure 4.7. Parasitic gate impedance in a MOS transistor with a bad aspect ratio [63]

The proposed solution is the use of multiple fingers (figure 4.8). The idea is to divide the transistor into several smaller transistors (fingers) connected in parallel. Using folding and interdigitating the parasitic capacitances, interconnect impedances and gate impedance are reduced [63]. The device share source/drain diffusions and have parallel gate impedances illustrated in figure 4.9.

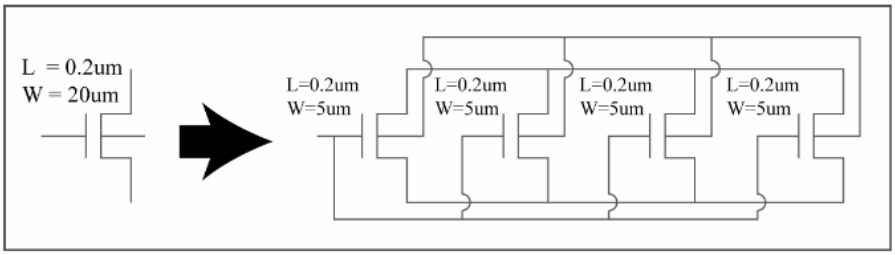

Figure 4.8. Dividing a single $(20\mu m/0.2\mu m)$ transistor into $4\,x\,(5\mu m/0.2\mu m)$ transistors using 4 fingers [63]

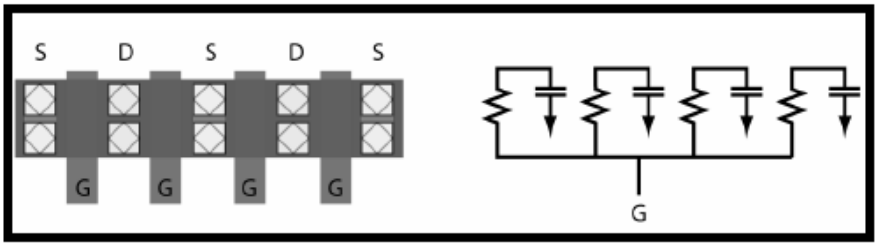

Figure 4.9. Parasitic gate impedance for a MOS transistor with the use of fingers [63]

As usual there is a tradeoff when altering the design. The tradeoff by using multiple fingers is reduced matching as the transistors become smaller. In general it is preferred to fold the transistor into an even number of fingers [63].

Figure 4.10 displays the difference in parasitic capacitances for; a bad aspect ratio, the use of 2 fingers and the use of 3 fingers. Here the positive effect of fingers is visible as the parasitic source-bulk and drain-bulk capacitances are reduced with $1/2$ and $1/3$ using 2 and 3 fingers respectively. When the parasitic capacitances are reduced the device speed increases.

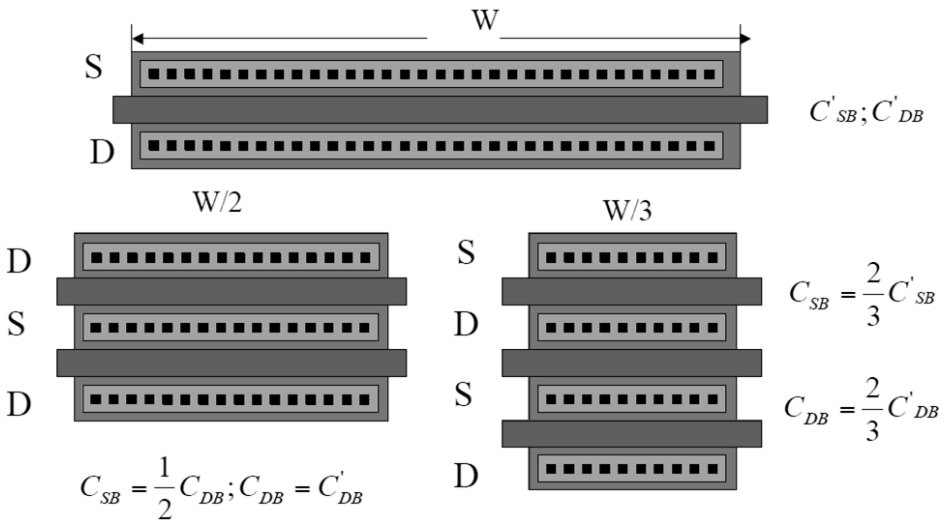

Figure 4.10. Use of multiple fingers and the reduction of parasitic capacitances [64]

The use of fingers on a transistor with a bad aspect ratio implemented in the design is displayed in figure 4.11. This is a small section of the current starved medium tune delay element's layout. In the medium tune delay elements the transistors are very wide compared to the length. The reason for this bad aspect ratio is to realize a high drain current which is required to achieve the desired high switching speed giving it high resolution. The transistor with the worst aspect ratio has a $\mathrm{W/L}$ ratio of $3\mu\mathrm{m}/0.1\mu\mathrm{m} = 30$. These bad aspect ratio transistors were divided into 2 fingers and folded, sharing source/drain diffusions. As seen from figure 4.11 the routing is a bit more complex using fingers as the output of the inverter suddenly is in-between the transistor gates. The use of multiple n-well, p-substrate, source and drain-contacts in addition to the use of fingers is also visible in figure 4.11.

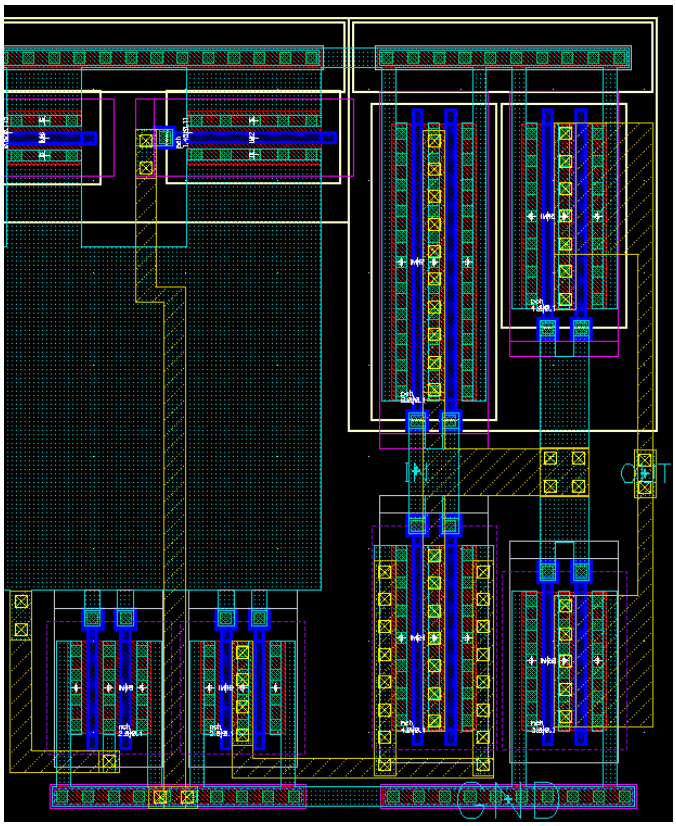

Figure 4.11. A small section of the current starved medium tune delay element, illustrating the use of fingers as well as multiple contacts

As the ending elements of interconnected transistors have different boundary conditions, dummies should be inserted to maintain equal conditions for the elements inside [64]. Dummies are shorted transistors and it is important to remember their parasitic contribution. Dummies can also be non-used cells like the dummies implemented in the coarse and medium tune elements.

These non-used cells are implemented to ensure equal load capacitance for all of the delay elements. More specifically it is inserted because the final (10^{th}) delay element has a smaller load than the others. It only has the multiplexer's input capacitance as load, making it faster than the others which also have the next cascaded delay element's input capacitance as load. A block diagram representing the implementation of the dummy in coarse tune is illustrated in figure 4.13.

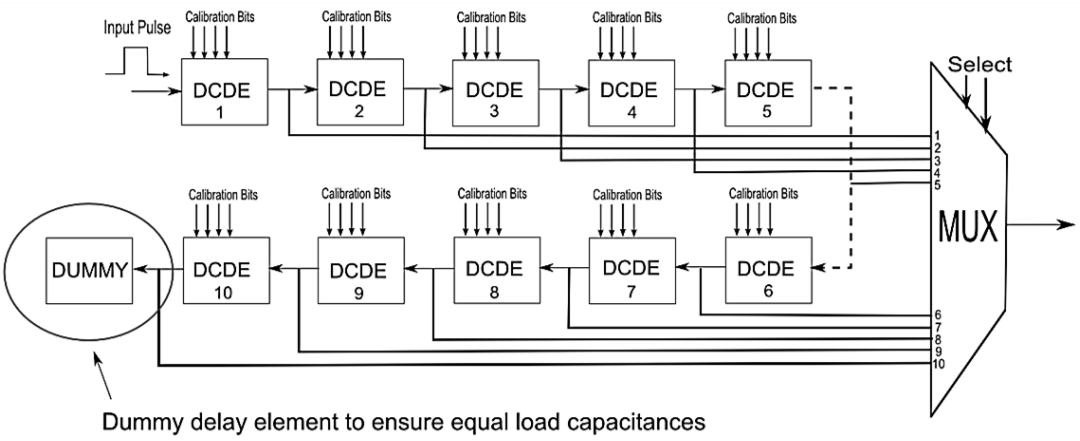

Figure 4.13. Implemented dummy cell in coarse tune to ensure equal load capacitances

Matching of single transistors and cells are important to achieve system performance. The shape, position and rotation of the devices make a difference. The shape (W and L) affect the device parameters as well as the rotation and position as Silicon is an un-isotropic material. The best way of ensuring a high degree of matching is to place the devices in the same direction and rotation with the same shape and preferably the same direction of current flow (figure 4.14).

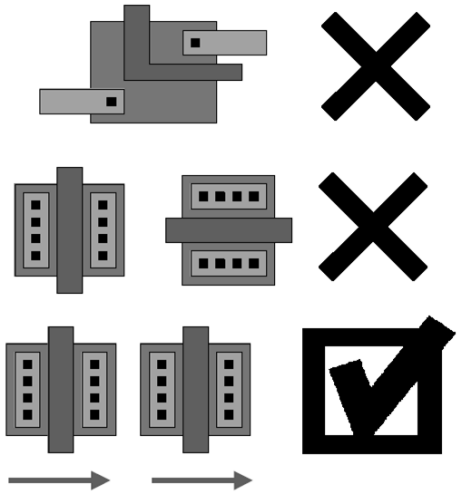

Figure 4.14 Matching of single transistors [64]

The matching of single transistors has been implemented in the design and can clearly be seen in the inverter layout in figure 4.15. Here the transistors have the same shape, are placed parallel to each other with the same rotation and direction and have the same current flow direction. The buffers are also matched by having a common axis of symmetry.

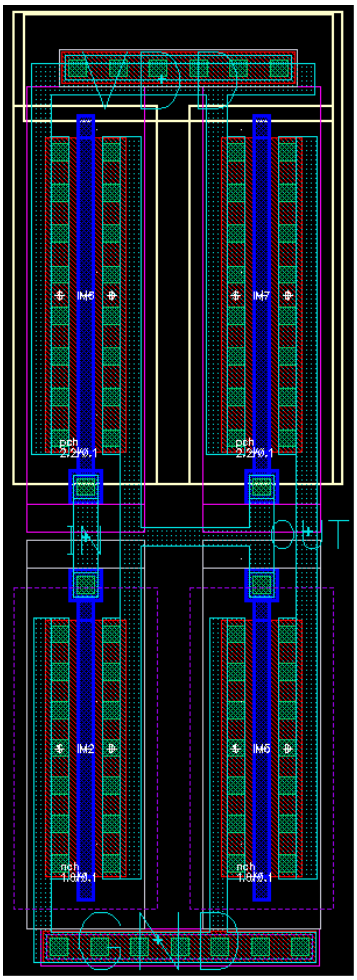

Figure 4.15. Matching of single transistors and devices using the common centroid pattern

To match two transistors (A and B) sharing a common node the axis of symmetry pattern or common centroid pattern should be considered.

Using the axis of symmetry pattern the two devices should be spitted into an equal number of fingers and then interdigitate the resulting number of elements. Figure 4.16 illustrates the axes of symmetry for 3 different approaches. In A and C the device has a common axis of symmetry which is preferred, whereas in B the device will be more affected by mismatch.

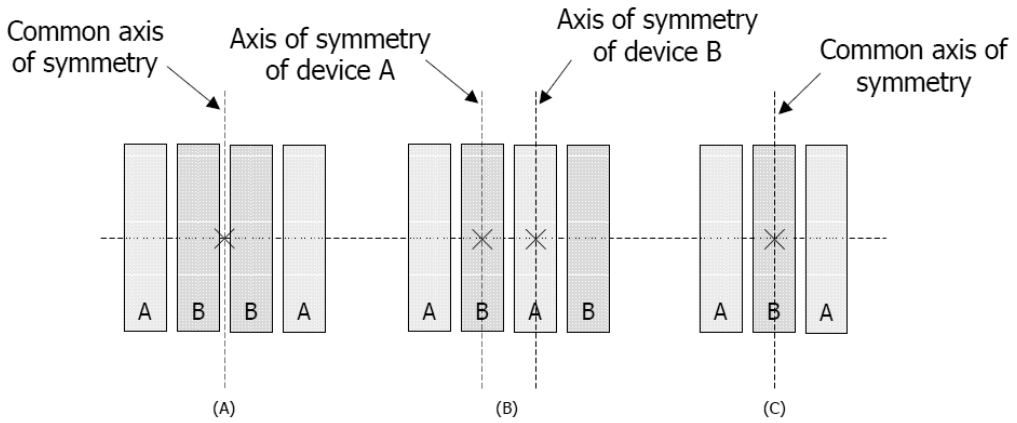

Figure 4.16. Axes of symmetry pattern for two devices using fingers [64]

Using the common centroid pattern the devices are oriented around a common centroid as shown in figure 4.17. After placing the first device DA the other devices are placed by mirroring the device around both axes giving rise to all configurations. The next step is to share source/drain diffusions and reduce parasitic capacitance if fingers are used. This configuration is the best for matching purposes [66] but the routing of metal and poly is more complex.

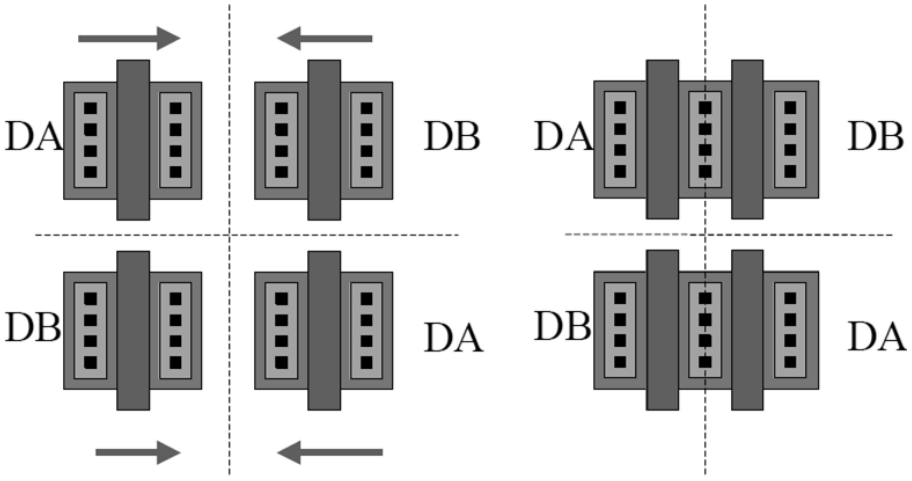

Figure 4.17. Common centroid pattern for two devices using fingers [64]

To ensure overall system matching, the signal paths for all equal devices have to be uniformly distributed as illustrated in figure 4.18. With high resolution comes great responsibility of matching the signal paths. A slightly longer signal path or higher impedance might result in significant delay mismatch, especially when they delays are as low as in the sub-one-hundred ps region. If the transistors are properly matched, the cells are matched and the signal path is uniformly distributed, overall system matching should be achieved.

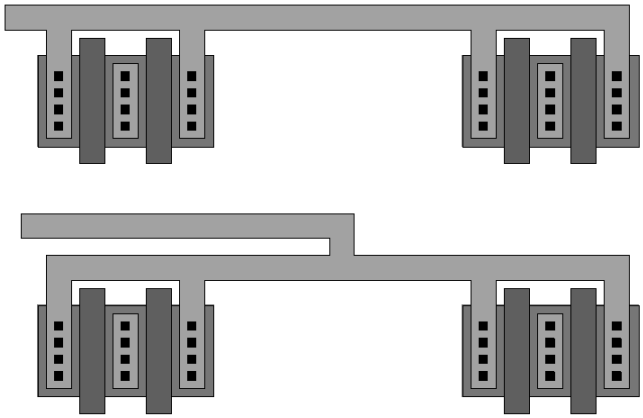

Figure 4.18. Uniformly distributing signal paths between two cells to achieve matched impedances [64]

In the implemented design uniform routing has been a design consideration, especially in the large multiplexers. Even though the tunable delay elements can compensate for these variations by tuning it is still preferable to consider and design a uniform distribution of the signal paths to achieve a well-balanced system with hopefully less variations to tune out. Then the tunability can rather be used to tune out other unwanted and non-designer related variations, such as temperature and process-variations (environmental variations).

Figure 4.19 illustrates the layout of the 2:1 multiplexer where the two signal paths has to be uniformly distributed to ensure equal delays for both inputs. It is important to match the signal paths and cells within the 2:1 MUX as this is the basis for creating the larger multiplexers. Matching the internal cells making up the multiplexer also makes it a lot easier to match the critical signal paths as only the routing connecting the cells together has to be considered at a higher level. The matched critical signal paths are marked in red to easier spot them.

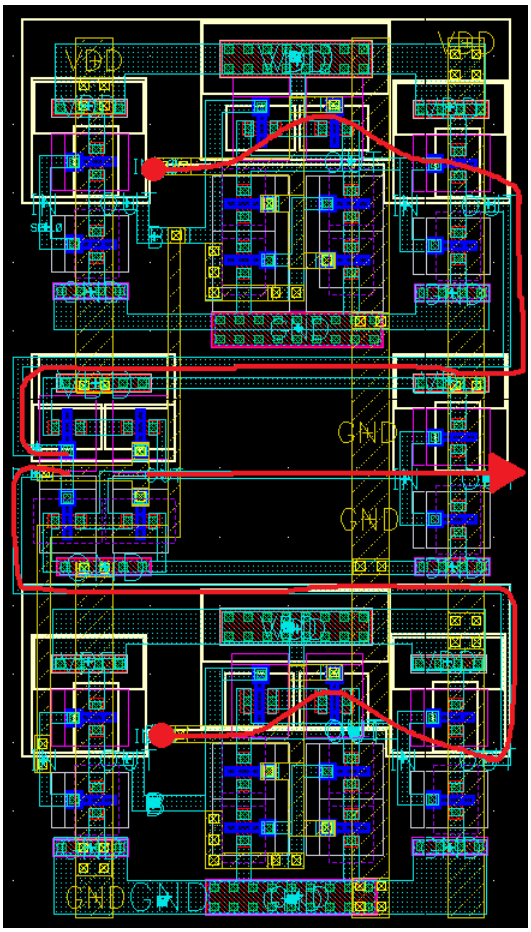

Figure 4.19. Uniformly distributed critical signal paths in the implemented 2:1 MUX layout

5 Test Bench

The test bench is another important part of the design to make sure the simulation set-up represents reasonable and real values giving rise to more accurate simulations related to the real world. Important parameters include:

- Load capacitances
- Input signal slew rate
- Post layout DUT parameters (PEX)
- Accuracy defaults

The test bench for the delay elements should incorporate a load capacitance equal to what the delay elements outputs actually see in the delay line. This means that the output is connected to another delay element in the cascaded delay line and to the parallel output multiplexer. That is the actual load capacitance for all of the delay elements. If the output is connected only to another delay element in the test bench the cells simulation will be inaccurate, e.g. too fast.

To simulate a reasonable input slew rate the input signal is propagated through two buffers. As the input slew rate will have an effect on the devices propagation delay and jitter, it is important to keep this at a reasonable level and not set the pulse generator to an unrealistic value, for instance a $1ps$ rise time.

Using post layout DUT parameters (PEX) gives a much more accurate simulation of the devices. Schematic simulations are usable and helpful while the cells are being designed, but post layout simulations and Monte Carlo simulations is a must to display the more realistic performance of the devices.

In the analog design environment (ADE) “spectre” there exist three different accuracy default settings depending on the desired accuracy of the simulations; conservative, moderate and liberal. For large designs lower accuracy settings (liberal) can be used to reduce the simulation time, but the high accuracy setting (conservative) should be used if possible.

By utilizing these techniques, accurate simulations close to the actual performance of the devices can be achieved. The “$tsmcN90rf$” library used to extract components (transistors, capacitors and resistors) for this design is well modeled, so the post layout simulations with the highest accuracy setting should exhibit close to “real-world” on-chip performance.

6 Results

Important test parameters include power consumption, average temperature dependency (ATD), process and mismatch-variations, INL, DNL, resolution, tunability and jitter. All of these parameters for all the different delay elements are compared and summarized in table 6.1 while table 6.2 and 6.3 summarizes the results for each delay line. The individual delay element and delay line results are presented in the succeeding chapters.

All post layout Monte Carlo simulations to simulate for process and mismatch-variations, temperature dependency, probability distribution and power consumption displayed in table 6.1 were performed with the fastest delay setting for the $DCDEs$ meaning a digital input vector of "0000". Simulations on the complete delay lines were schematic simulations, not post layout simulations due to the extremely long processing time, all of which were performed using the highest accuracy setting (conservative).

PSS and Pnoise spectre simulations were performed to simulate periodic cycle-to-cycle jitter, where cycle-to-cycle jitter is the difference in length/duration of any two adjacent clock periods. The beat frequency (clock frequency) used was the desired PRF of $100MHz$, the temperature was set to 27℃ and 10 harmonics were included in the noise simulations.

To determine the linearity of the delay elements, a straight-line INL and DNL were calculated using the post layout simulation results at 27℃. A best-fit line could be used to lower the INL and DNL results but the comparison is the main interest, not to achieve the lowest value possible by altering the calculation technique. Straight-line INL and DNL are calculated using equation 6.1 and 6.2 respectively [67]:

$$INL = \left(\frac{D_i - D_0}{D_{LSB}}\right) - i, \text{ where } 0 < i < 2^i - 1 \quad (6.1)$$

where D_i is the delay at index i, D_0 the delay value at index 0 and D_{LSB} is the ideal delay value of the least significant bit.

$$DNL = \left(\frac{D_{i+1} - D_i}{D_{LSB}}\right) - 1, \text{ where } 0 < i < 2^i - 2 \quad (6.2)$$

where D_{i+1} is the delay value at index $i + 1$, D_i is the delay value at index i and D_{LSB} is the ideal delay value of the least significant bit.

Delay Element	Avg PWR μW	ATD $ps/℃$	Mismatch % $within\ 1\sigma$	INL LSB	DNL LSB	Resolution ps	Jitter CC ps rms	Silicon Area μm^2
CSCT	272	0.103	70.4	4.57	1.86	-	1.280	683.06
SCCT	245	0.119	72.2	0.99	0.37	-	0.364	856.73
BCT	371	0.111	72.7	-	-	-	0.285	621.03
CSMT	699	0.099	78.0	3.08	0.88	67.71	0.206	104.06
SCMT	74	0.167	74.0	0.96	0.49	54.42	0.102	109.98
BMT	65	0.118	72.0	-	-	27.50	0.042	15.30

Table 6.1. Post layout comparison parameters for the different delay element architectures

From table 6.1 all of the comparison parameters for all the delay elements in both medium and coarse-tune can be seen. Post layout temperature dependency for the coarse and medium-tune delay elements are illustrated in figure 6.1 and 6.2 respectively. It is not as easy to spot the differences between the coarse tune delay elements but in the plot of the medium tune delay elements it is visible that the current starved coarse tune ($CSCT$) architecture actually has the lowest temperature dependency followed by the buffer coarse tune (BCT) and the shunt capacitor coarse tune ($SCCT$).

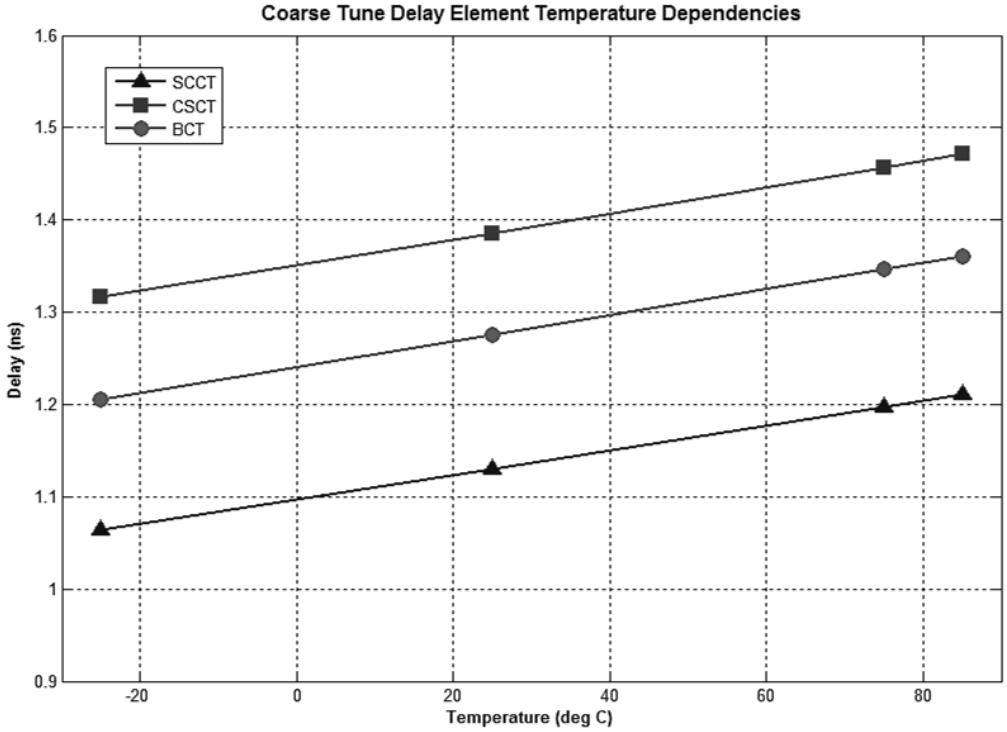

Figure 6.1. Post layout temperature dependency for all coarse tune delay elements

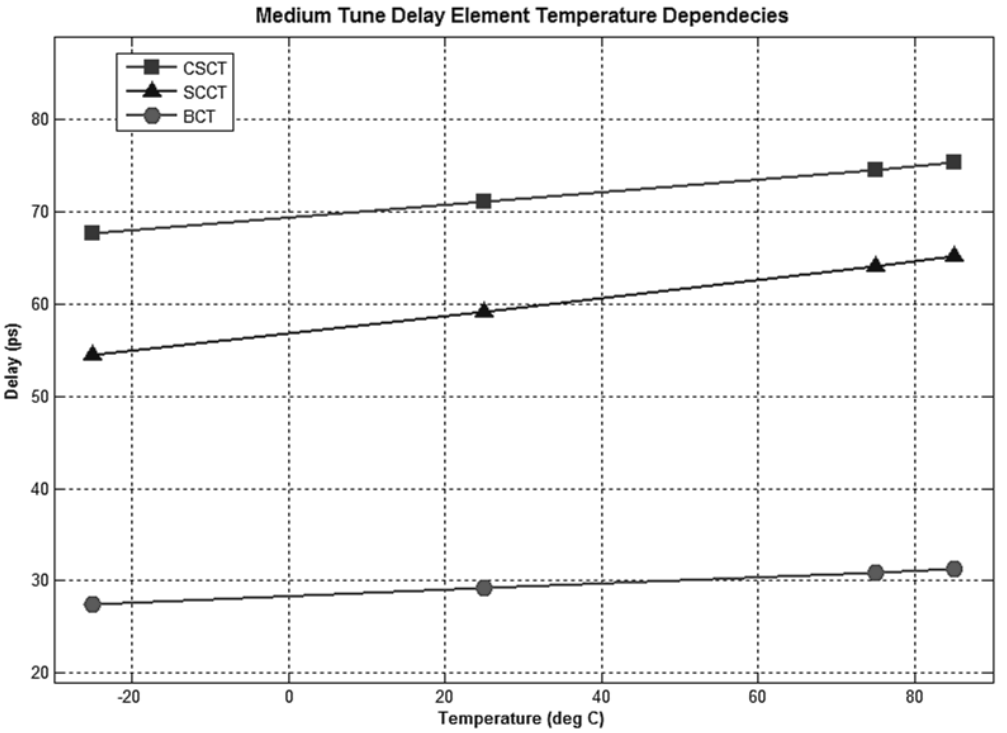

Figure 6.2. Post layout temperature dependency for all medium tune delay elements

The results presented in table 6.2 and 6.3 compares the different delay line architectures. For each delay line there is a "$Fast$" and a "$Slow$" parameter. The fast parameter means that the delay elements are tapped at the first tap in both coarse and medium-tune achieving the lowest possible delay. For the $DCDEs$ (delay line 1 and 2) the fast parameter also means that each delay element is set to the fastest setting ("0000") using the calibration bits.

The slow parameter is the opposite; the delay elements are tapped at the last tap in both coarse and medium-tune to achieve the highest delay. Now the $DCDEs$ are set to the slowest calibration setting ("1111") giving them extra tunability over the non-tunable buffers.

Viewing the results in table 6.2 it is clear that the current starved delay line suffers greatly from pulse width reduction due to asymmetry on the slow settings as well as having the highest power consumption. In table 6.3 on the other hand it exhibits the lowest temperature dependency. The introduction of tunable delay elements clearly result in tradeoffs that have to be taken into consideration when choosing the design architecture.

Delay Line		Delay @ 27℃ ns	Avg PWR mW	Jitter CC $ps\ rms$	Pulse width % $of\ clock$	Tunable Range ns
1 ***CS***	**Fast**	1.88	13.30	1.52	98.0	11.89
	Slow	13.77	8.14	-	8.3	
2 ***SC***	**Fast**	1.91	0.65	0.894	104.8	14.31
	Slow	16.22	0.66	-	106.4	
3 ***NPB***	**Fast**	1.88	0.86	0.743	105.9	10.41
	Slow	12.29	1.05	-	102.0	

Table 6.2. Comparison parameters for the different delay lines

Delay Line		Delay @ −25℃ ns	Delay @ 85℃ ns	% of change
1 ***CS***	**Fast**	1.67	1.95	16.77
	Slow	12.99	14.66	12.86
2 ***SC***	**Fast**	1.78	2.14	20.22
	Slow	14.97	17.69	18.17
3 ***NPB***	**Fast**	1.74	2.04	17.24
	Slow	11.51	13.16	14.34

Table 6.3. Temperature dependency comparison of the different delay lines

6.1 Current Starved Coarse Tune Delay Element

A comparison of the original current starved coarse tune delay element and the improved version is shown in figure 6.3 and 6.4. The effects of the improvements are clearly visible. The slew rate is drastically improved as well as the pulse width reduction. The worst case rise time, meaning the rise time of the first buffers output at the slowest setting, has been improved from $278ps$ to a quite fast $98ps$ that reduces the jitter noise. In the original design the pulse width is almost halved with the slowest setting, meaning that after two cascaded delay elements the input pulse would be gone.

Intrinsic delay and tunability of the improved delay element are shown in figure 6.4 and 6.5.

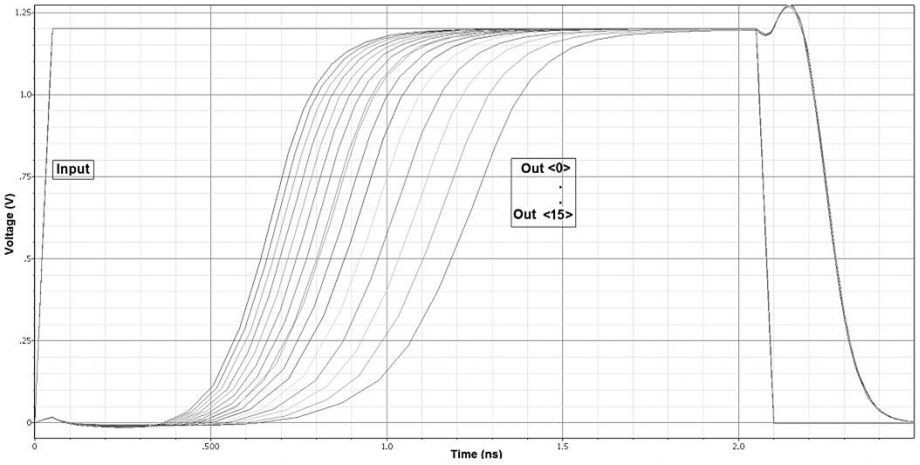

Figure 6.3. Post layout simulation of all input vectors for the original current starved coarse tune delay element

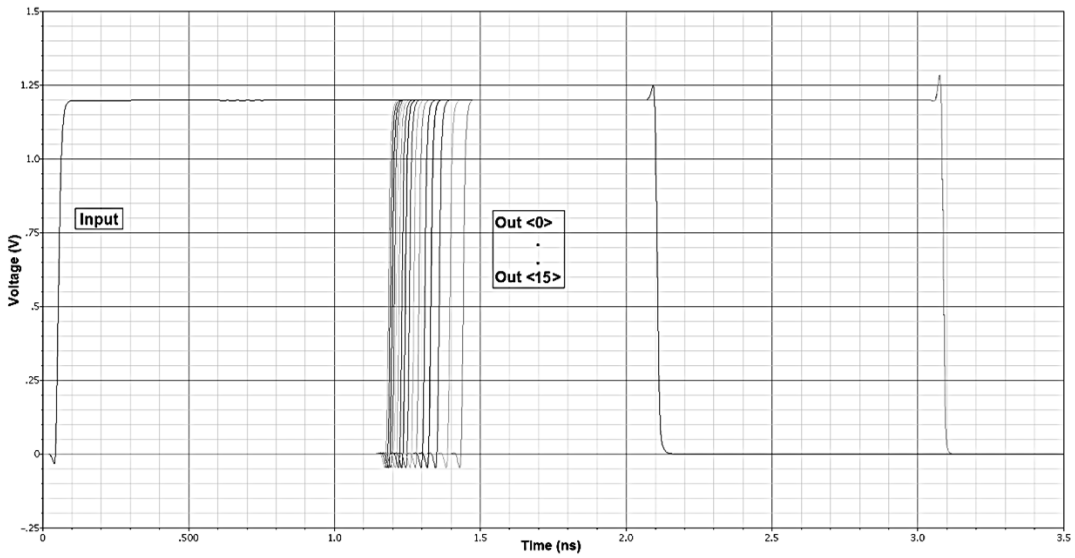

Figure 6.4. Post layout simulation of all input vectors for the improved current starved coarse tune delay element

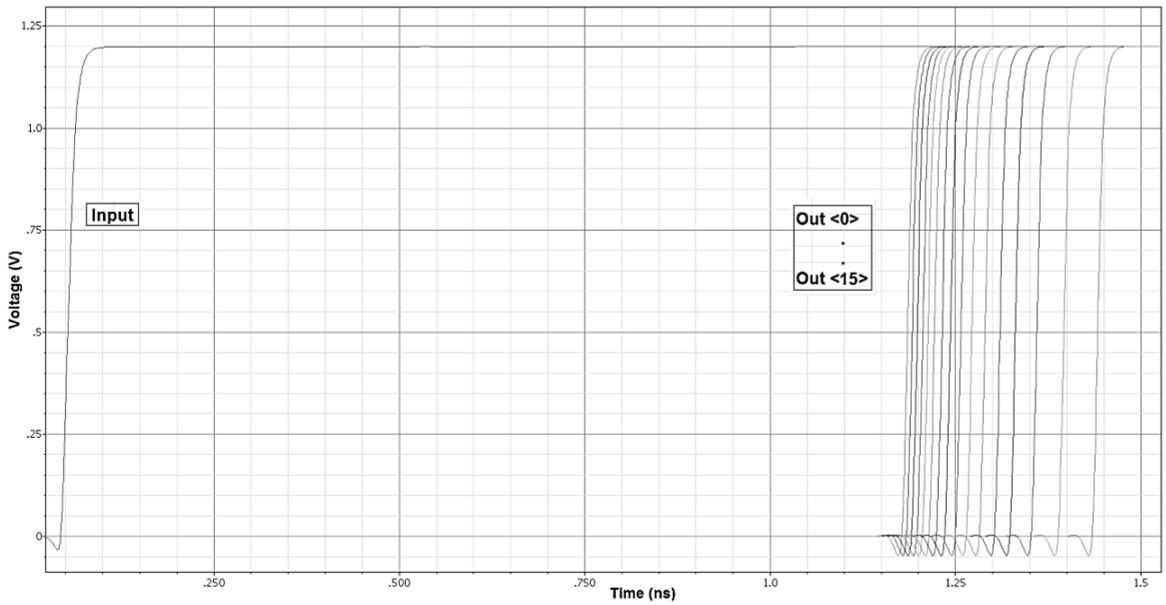

Figure 6.5. Post layout simulation of all input vectors for the improved current starved coarse tune delay element 2

Compared to the proposed delay element in [1] the implemented design has drastically improved the slew rate, thus reduced jitter noise, without sacrificing very much temperature dependency by introducing the extra buffers. The implemented and improved architecture exhibits a temperature change of 5.18% compared to 3.5% over a temperature range of $25 \rightarrow 75℃$. It is also to be noted that the implemented design achieves this having 2 times the intrinsic delay.

Figure 6.6 depicts the linearity of the delay element as well as post layout delay versus schematic delay. Note that the intrinsic post layout delay has increased with around 20% from the schematic due to routing resistance and capacitance as well as other parasites. The non-linearity of the current source is also clearly visible.

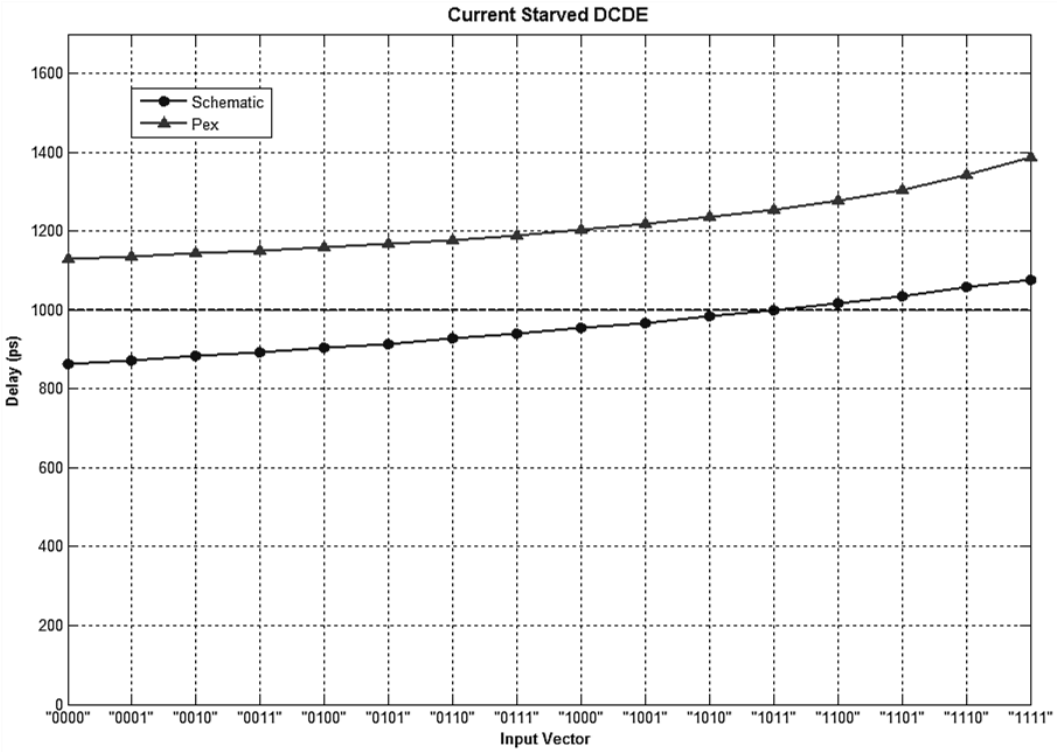

Figure 6.6. Post layout versus schematic delay and linearity for the improved coarse tune current starved delay element

Monte Carlo simulations displaying process and mismatch-variations as well as temperature dependency is displayed in figure 6.7. The temperature dependency is very good, actually the best, even lower than the non-tunable buffers. Probability distribution of the current starved delay element is shown in figure 6.8. 70.4% of the simulations are within 1σ which is the lowest percentage of the three coarse tune delay elements, even though it was quite close to the others. It is to be noted that the probability distribution should follow a Gaussian distribution as explained in chapter 2. The reason why the results differ is that the number of Monte Carlo simulation runs was not high enough for the coarse tune delay elements due to very long simulation times and disc space usage. If a higher number of runs were performed the results should be distributed like a Gaussian distribution.

The power consumption of the delay element under switching is illustrated in figure 6.9. Here the pulse traveling through all the buffers inside the delay element is visible.

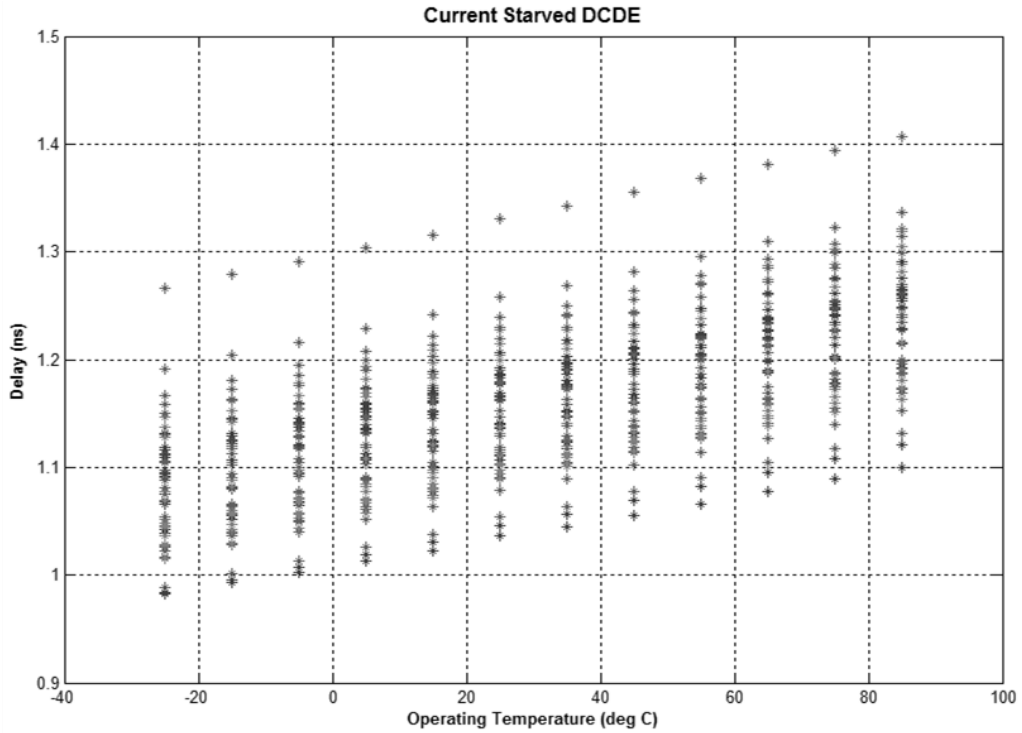

Figure 6.7. Post layout Monte Carlo simulations for the improved current staved coarse tune delay element

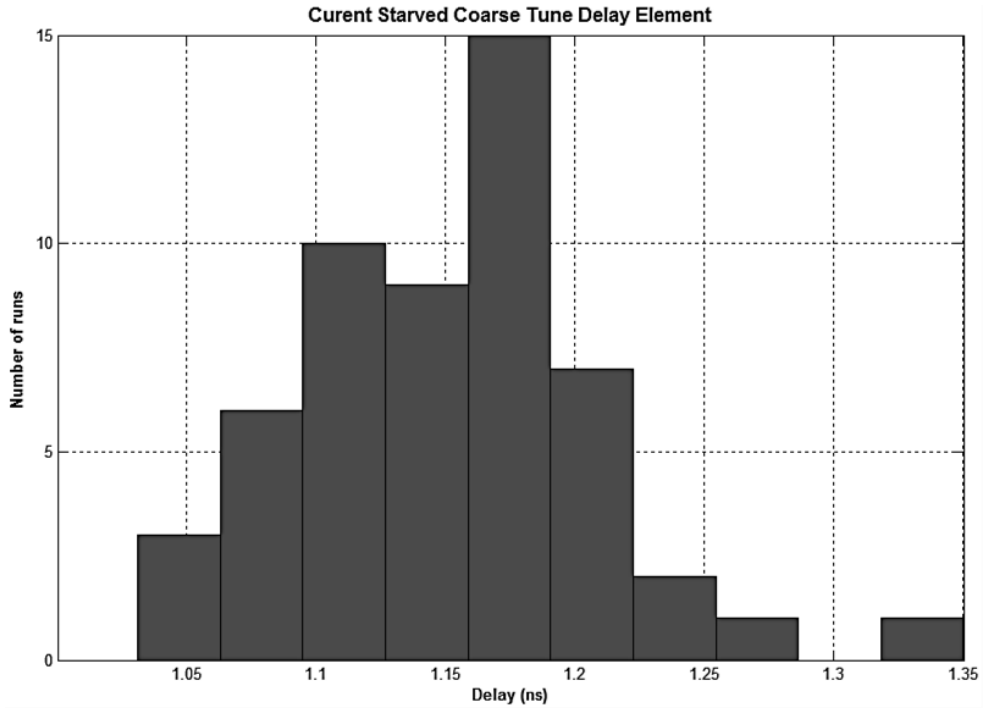

Figure 6.8. Post layout probability distribution for the improved current starved coarse tune delay element

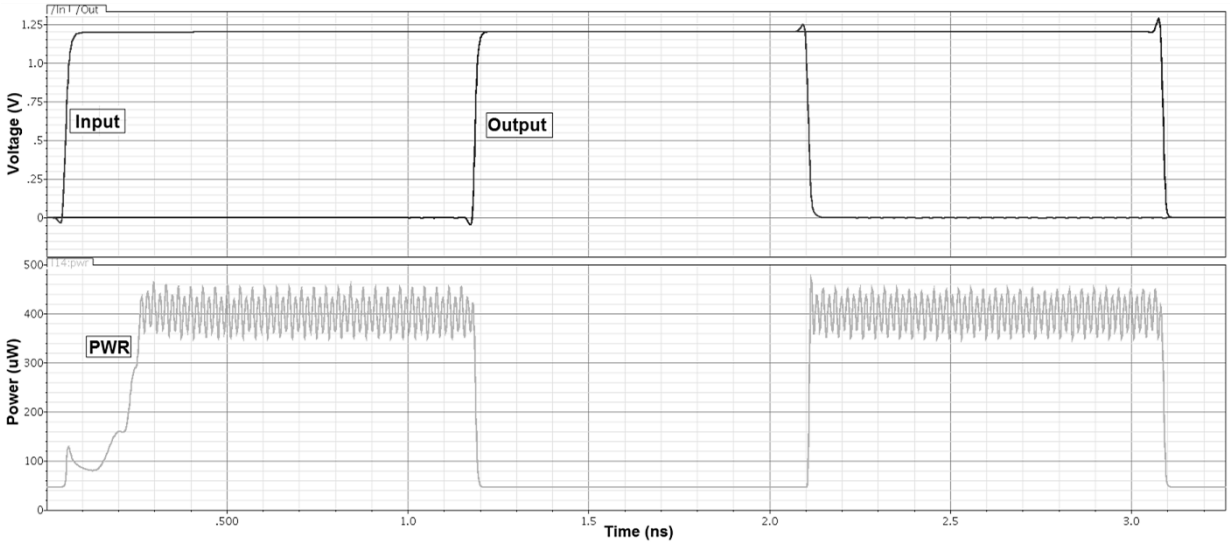

Figure 6.9. Post layout power consumption for the current starved coarse tune delay element

The improved current starved delay element (figure 3.8) used in this design only controls the rising edge of the signal and as mentioned this results in pulse width reduction due to asymmetry. The pulse width is reduced by approximately $90ps$ as it passes each delay element relative to the input signal (figure 6.10).
In figure 6.10 the input signal pulse width is $2ns$ and it is not hard to see that after a certain number of delay elements the pulse passes through the signal will disappear. This limits the maximum PRF and the maximum number of cascaded delay elements of the circuit. By using the desired $100MHz$ PRF the pulse width is $5ns$ for a 50% duty cycle meaning that the pulse could travel through 50 delay elements before vanishing.

To avoid this from happening there are at least two solutions. The first is a pulse shaper, which transforms the pulse back to a certain pulse width, which can be inserted after a certain number of delay elements the pulse passes through. Where the pulse shaper is inserted can be calculated depending on the input pulse width or PRF and the number of cascaded delay elements.

The other solution is to modify the delay elements to control the rising *and* falling edge of the pulse, making it symmetric. It is to be noted that if the input pulse can be wide enough and only the rising edge of the signal is of interest the pulse width reduction will not be a problem.

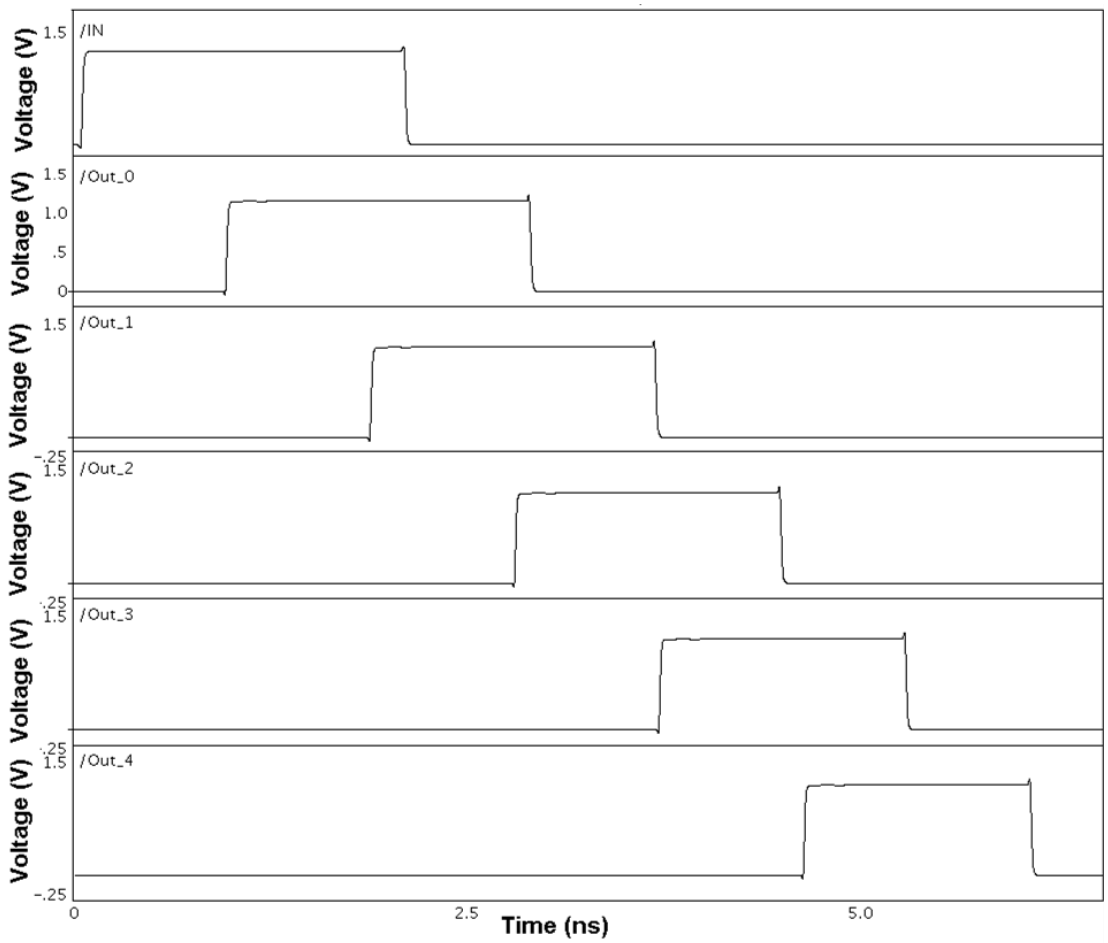

Figure 6.10. Post layout pulse width reduction as a result of asymmetric behavior in the current starved coarse tune delay element

The cycle-to-cycle jitter noise of the current starved coarse tune delay element, as seen from table 6.1, is the worst of the group by a fair amount with $1.28ps\ rms$. This is a good example of that there always is a tradeoff to take into account as the delay element has the definitively lowest temperature dependency but as a result the highest jitter.

6.2 Shunt Capacitor Coarse Tune Delay Element

A comparison of the original shunt capacitor coarse tune delay element and the improved version is represented by figure 6.11 and 6.12. It can be seen that the slew rate has been drastically improved. The worst case rise time, meaning the rise time of the first buffers output at the slowest setting, has been improved from $372ps$ to a much faster $58ps$, reducing the jitter noise.

Intrinsic delay and tunability of the improved delay element is displayed in figure 6.12 and 6.13.

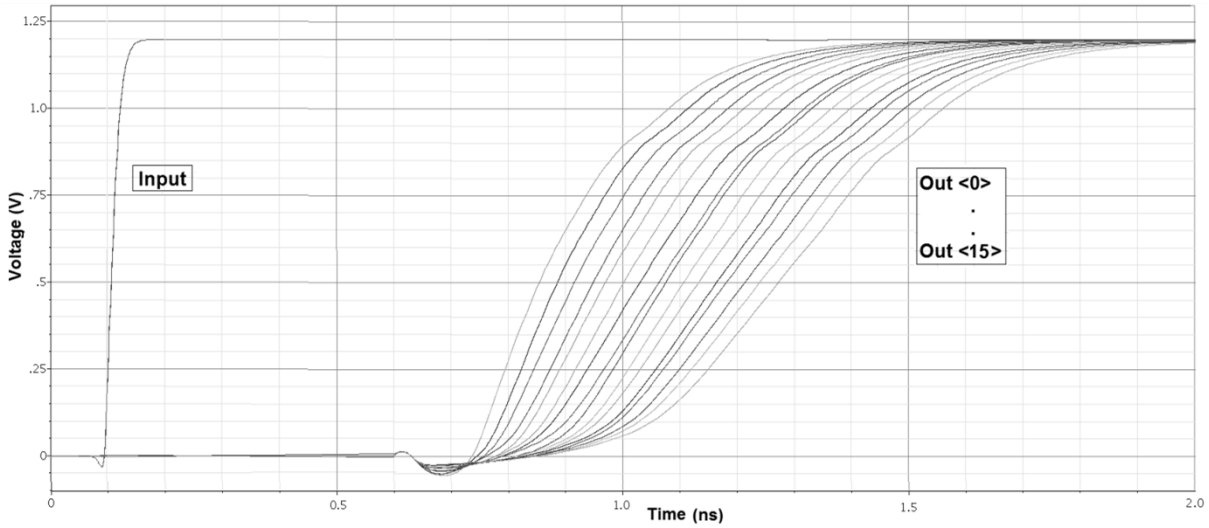

Figure 6.11. Post layout simulation of all input vectors for the original shunt capacitor coarse tune delay element

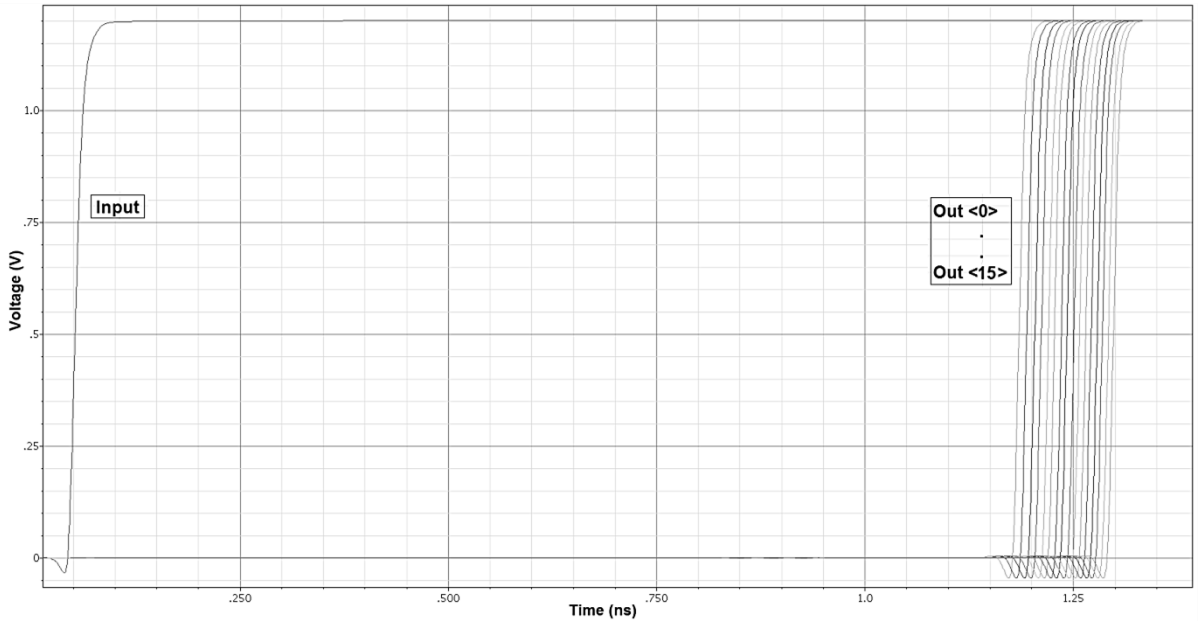

Figure 6.12. Post layout simulation of all input vectors for the improved shunt capacitor coarse tune delay element

Figure 6.12 and 6.13 are post layout simulations of the delay elements tunability. It is visually noticeable that the delay element exhibits quite good linearity, especially compared to the current starved coarse tuned delay element and that it does not suffer from any pulse width reduction as both the rising and falling edge of the signal is adjusted, making it symmetric.

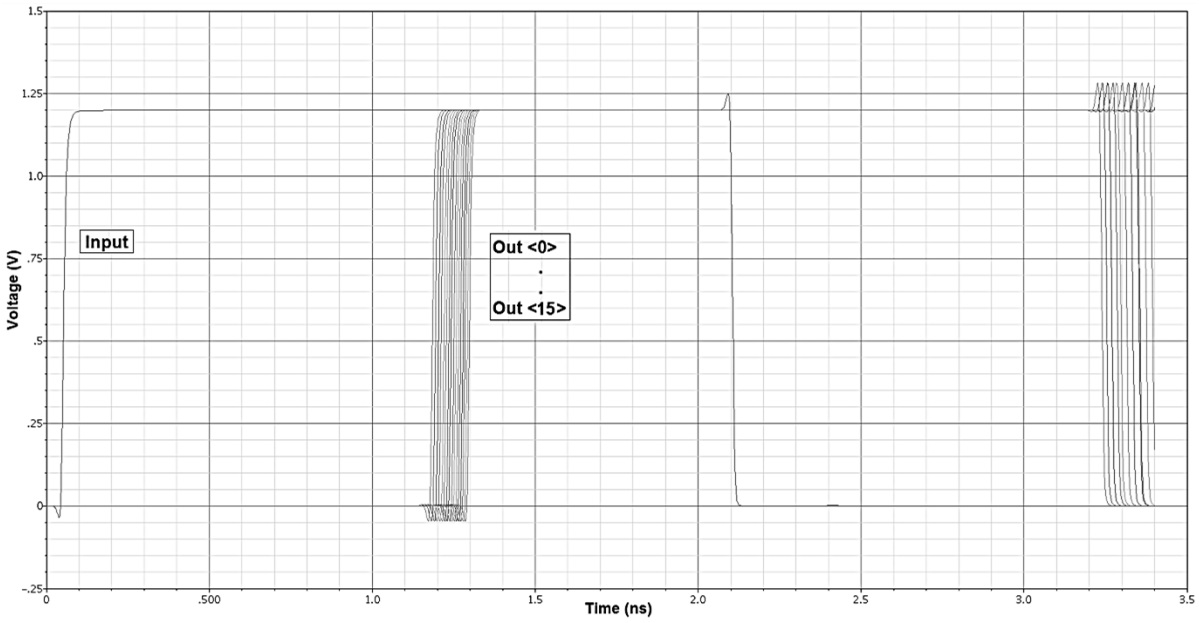

Figure 6.13. Post layout simulation of all input vectors for the improved shunt capacitor coarse tune delay element 2

In figure 6.14 the linearity of the delay element as well as post layout versus schematic delay is displayed. Here the delay has increased with approximately 20% in post layout simulations as the routing and parasites are included. Note that the delay element actually became more linear in post layout simulations.

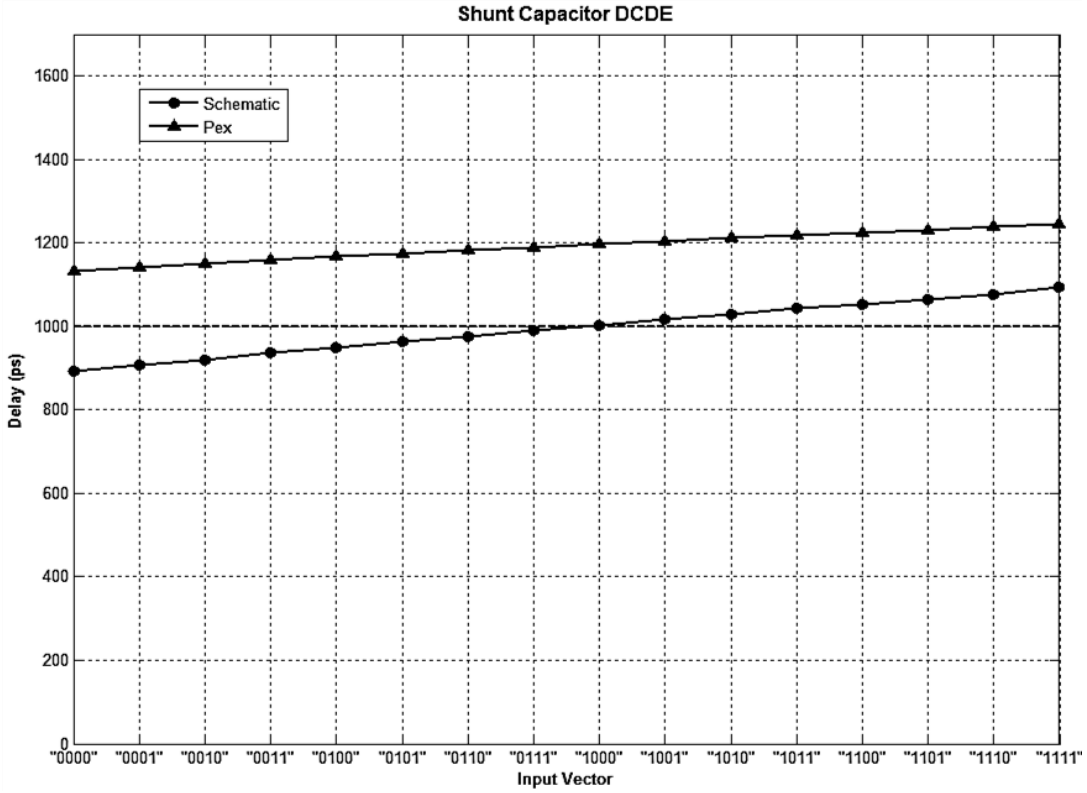

Figure 6.14. Post layout versus schematic delay and linearity for the shunt capacitor coarse tune delay element

A post layout Monte Carlo simulation with process and mismatch-variations as well as temperature dependency is shown in figure 6.15. The shunt capacitor delay element is clearly more temperature dependent than the current staved architecture, but has lower process and mismatch-variations.

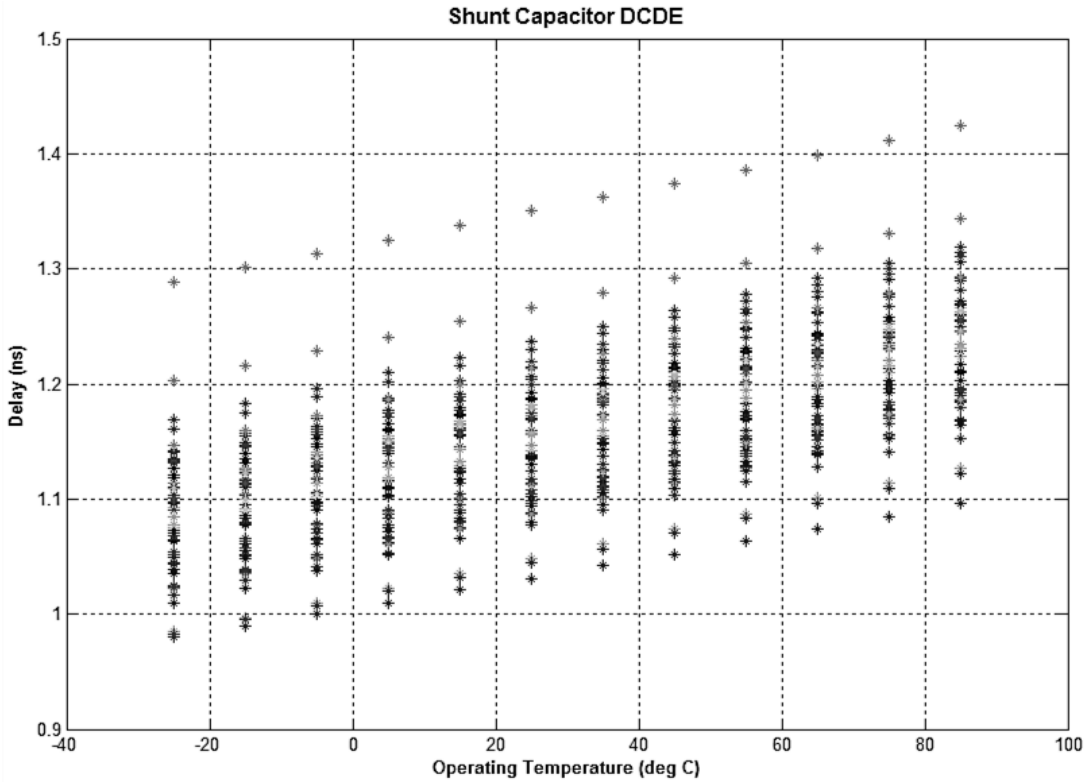

Figure 6.15. Post layout Monte Carlo simulations for the shunt capacitor coarse tune delay element

Probability distribution and power consumption is finally illustrated in figure 6.16 and 6.17. Regarding the probability distribution 72.2% of the simulations are within 1σ which is the next best of the three architectures. The distribution, as mentioned for the current starved coarse tune, does not quite follow a Gaussian distribution due to not enough simulation runs.

The average power consumption of the shunt capacitor coarse tune delay element is actually the lowest in the coarse tune group. The pulse traveling through all the buffers inside the delay element is visible in the power consumption plot (figure 6.17).

Even though the shunt capacitor coarse tune delay element has a higher temperature dependency than the current starved architecture it has significantly lower jitter, as seen in table 6.1. With a cycle-to-cycle jitter of $0.364ps\ rms$ it compares quite well to the benchmark, the non-tunable buffers, which has $0.285ps\ rms$.

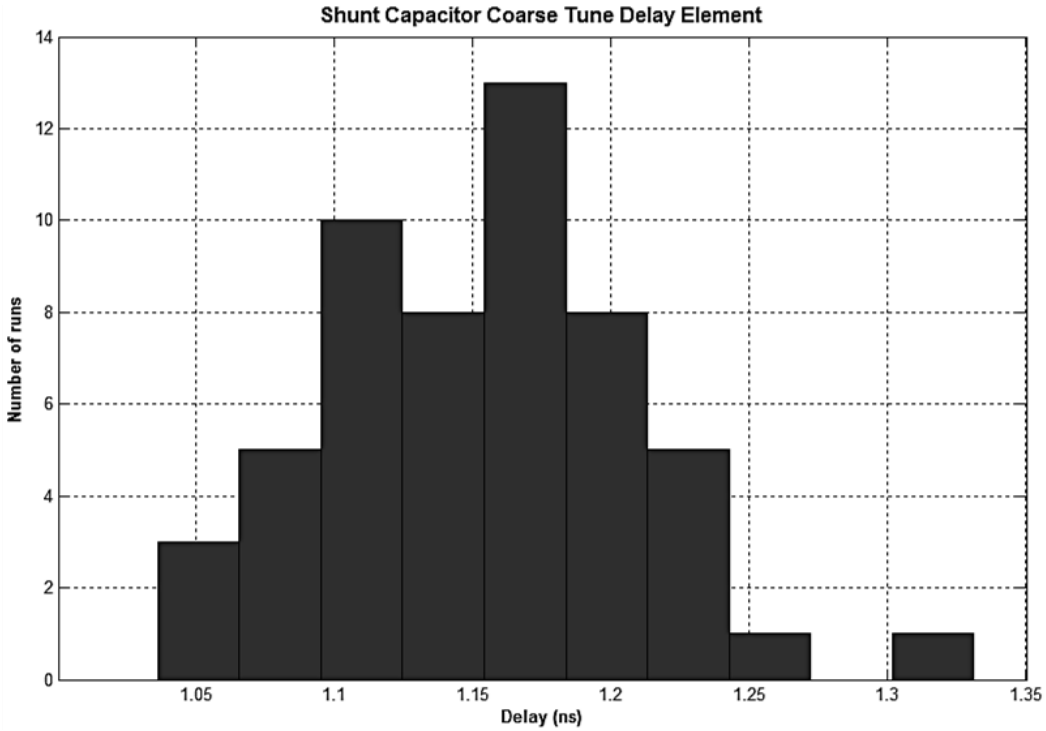

Figure 6.16. Post layout probability distribution for the shunt capacitor coarse tune delay element

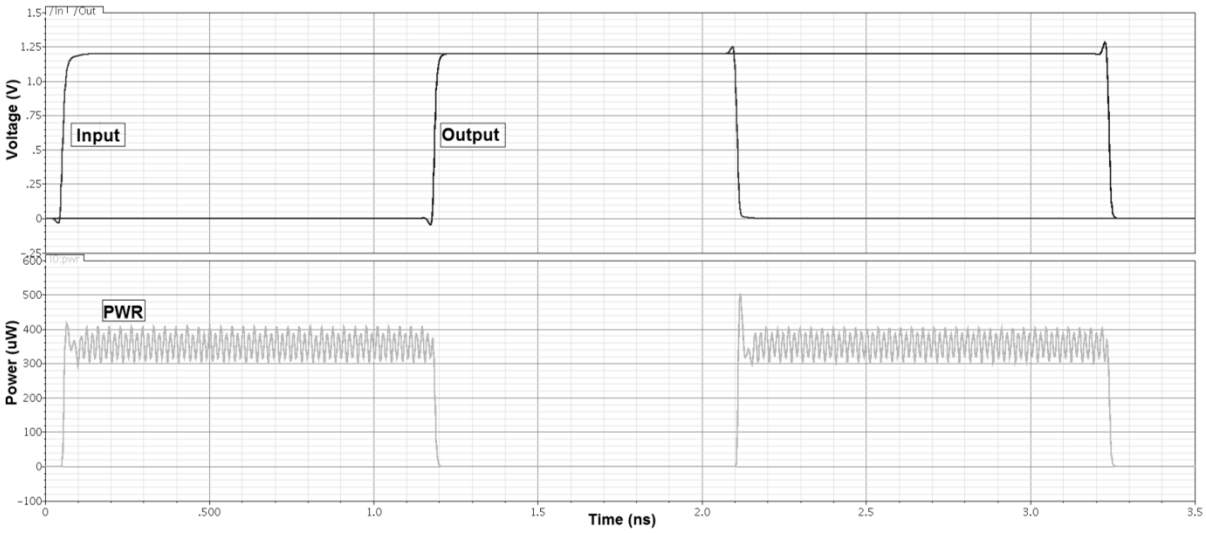

Figure 6.17. Post layout power consumption for the shunt capacitor coarse tune delay element

6.3 Buffer Coarse Tune Delay Element

The buffer coarse tune delay element has no intrinsic tuning but figure 6.18 shows a post layout simulation displaying the $1ns$ intrinsic delay. It is to be noted that also here the actual delay is approximately 20% higher than the schematic delay due to post layout extracted parasites and routing delay considerations.

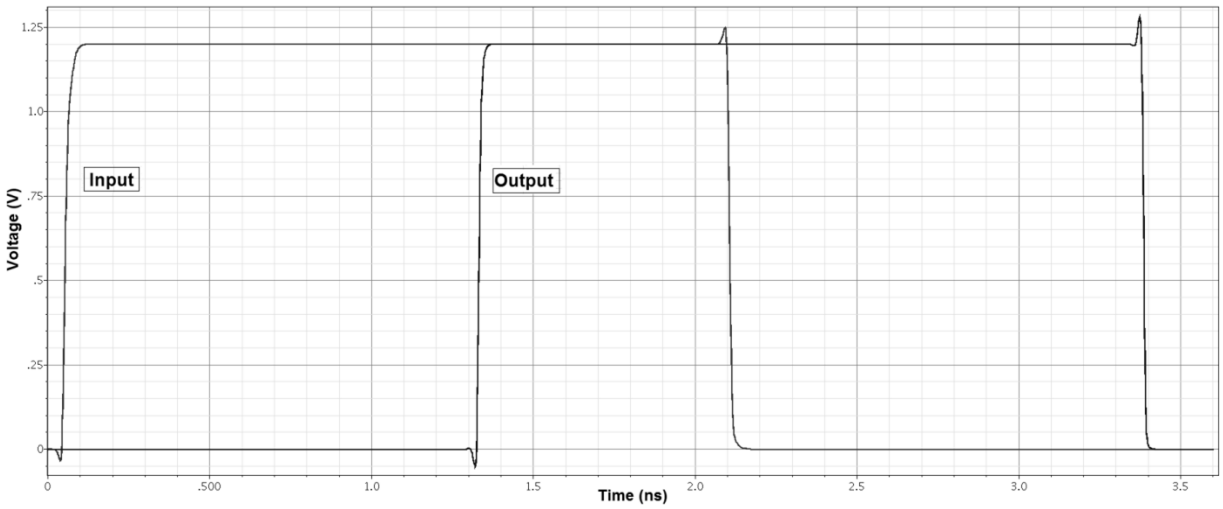

Figure 6.18. Post layout simulation of intrinsic delay for the buffer coarse tune delay element

In figure 6.19 the Monte Carlo simulations of process and mismatch-variations are displayed. The buffer coarse tune delay element has the least process and mismatch-variations of the coarse tune delay element architectures with 72.7% of the simulations within 1σ. The temperature dependency of the delay element is actually not the best in the group as it is beaten by the current starved coarse tune delay element.

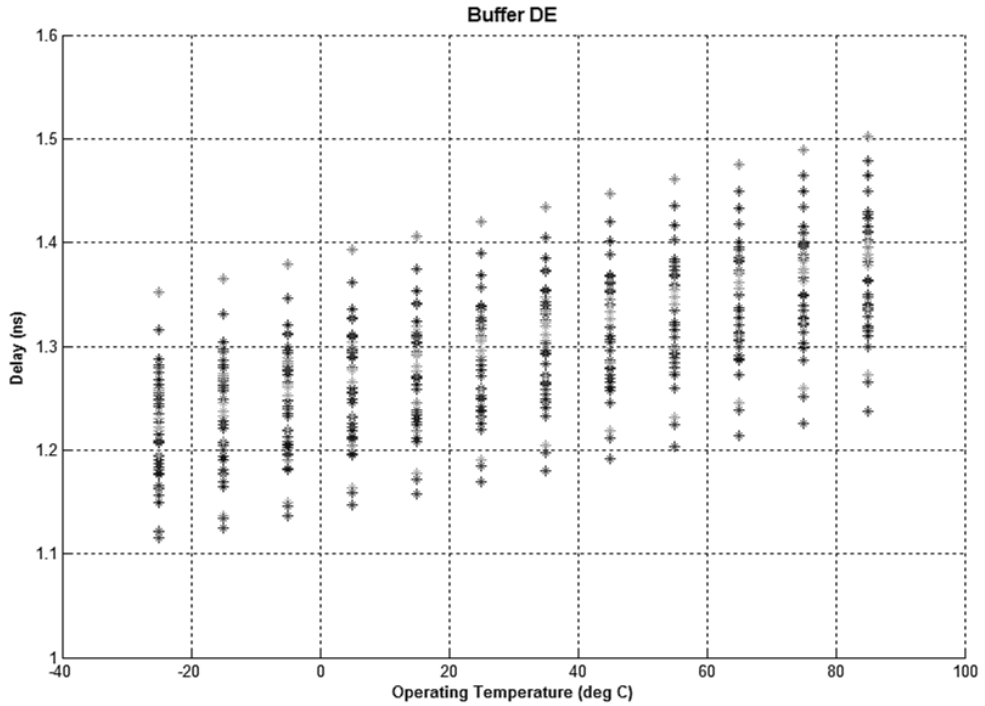

Figure 6.19. Post layout Monte Carlo simulations for the buffer coarse tune delay element

From figure 6.20 and 6.21 the probability distribution and power consumption can be seen respectively. It is noticeable that the buffer coarse tune delay element has the highest power consumption, by a fair amount, compared with the two other coarse tune architectures. Considering that it consists of more and faster delay elements this becomes quite logical.

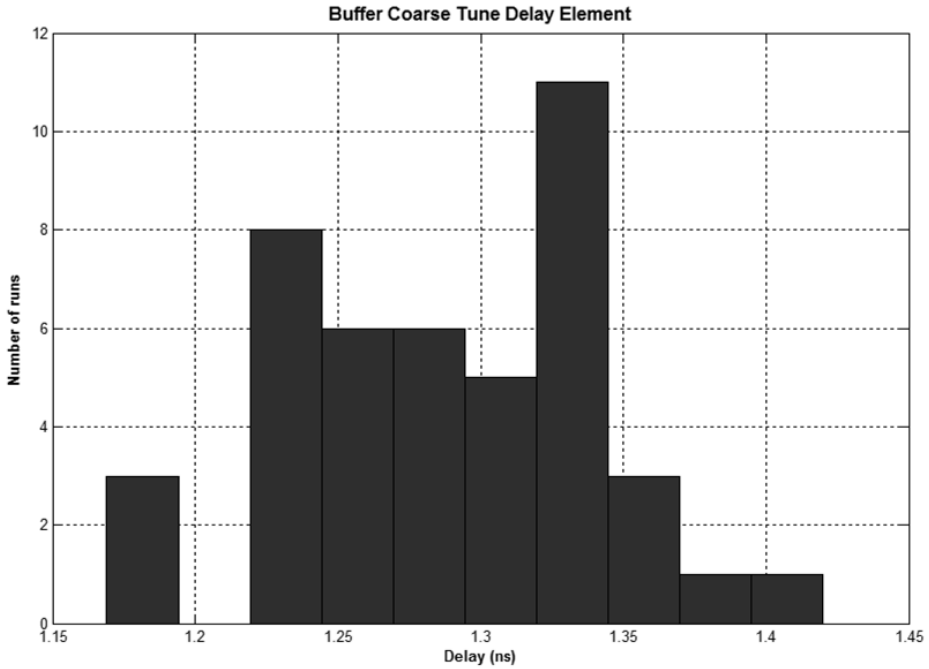

Figure 6.20. Post layout probability distribution for the buffer coarse tune delay element

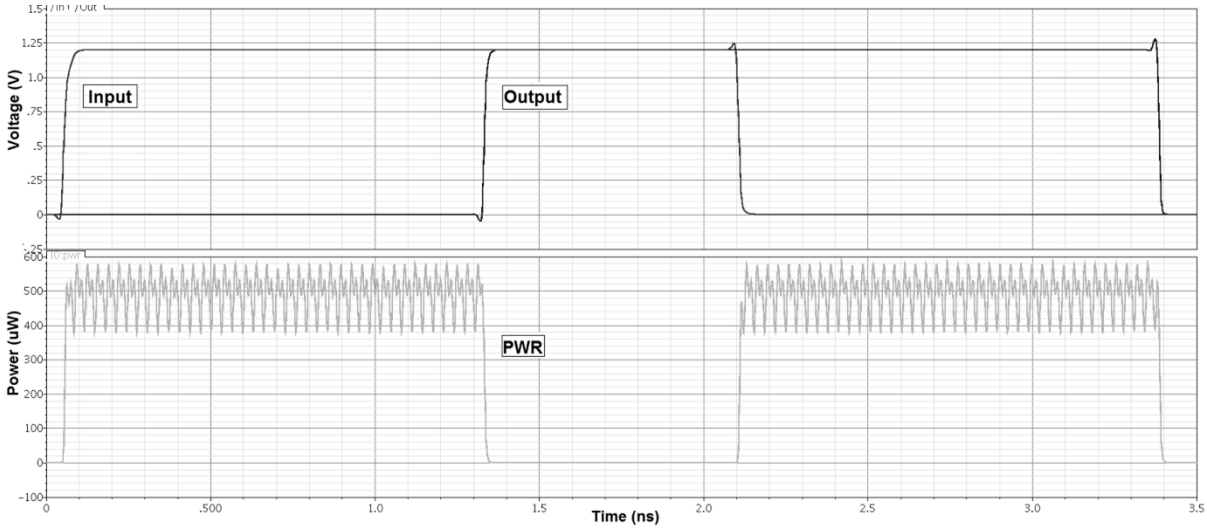

Figure 6.21. Post layout power consumption for the buffer coarse tune delay element

The non-programmable buffers were assumed to be the benchmark for jitter noise and this proves to be correct by viewing table 6.1 The buffers clearly represent the lowest cycle-to-cycle jitter with $0.285ps\ rms$ compared to the shunt capacitor's $0.364ps\ rms$ and the current starved's $1.28ps\ rms$.

6.4 Current Starved Medium Tune Delay Element

Figure 6.22 and 6.23 displays the post layout tunability of the delay element with the 16 different input vector settings. Here, as for the current starved coarse tune delay element, the non-linearity of the current source can be seen even though it is less dominating in the medium tune delay element due to less current starving. Note that the test bench input pulse seems to have a slightly low slew rate even though it has been passed through 2 buffers.

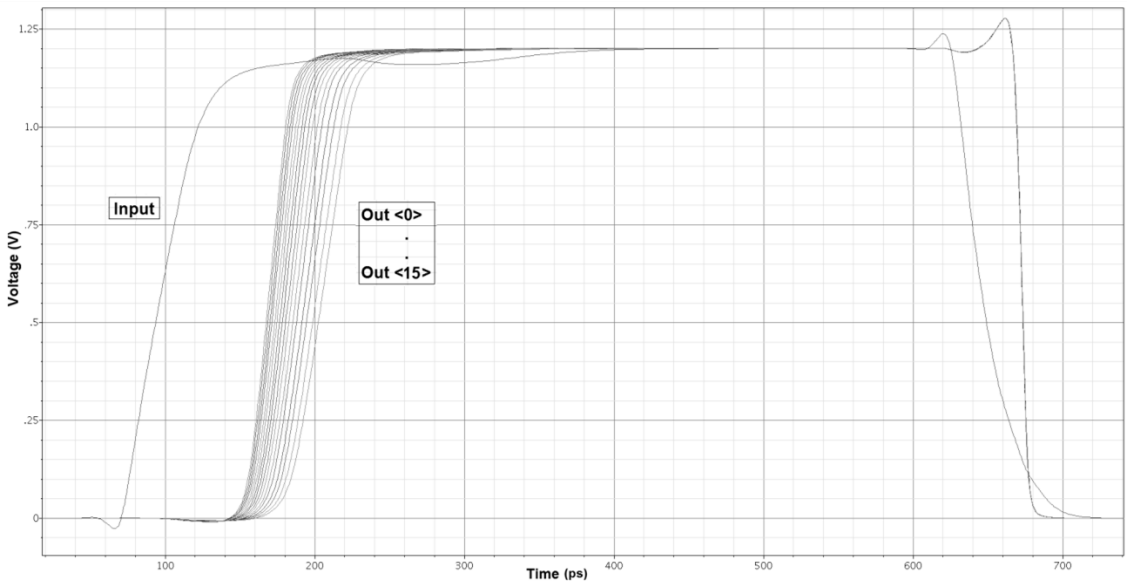

Figure 6.22. Post layout simulation of all input vectors for the current starved medium tune delay element

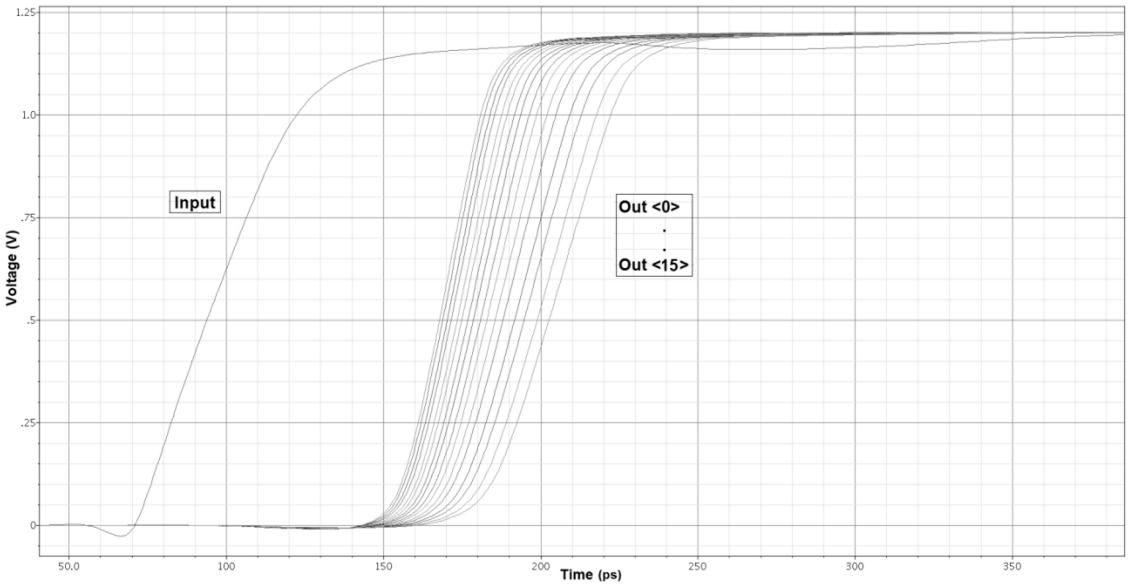

Figure 6.23. Post layout simulation of all input vectors for the current starved medium tune delay element 2

The linearity and post layout versus schematic delay can be seen in figure 6.24. Intrinsic delay has increased in post layout due to extracted parasites and routing delays. Note that the medium tune intrinsic delay has increased with almost 40% compared to the coarse tune's 20% due to the fact that the routing has a higher influence relative to the delay elements intrinsic delay.

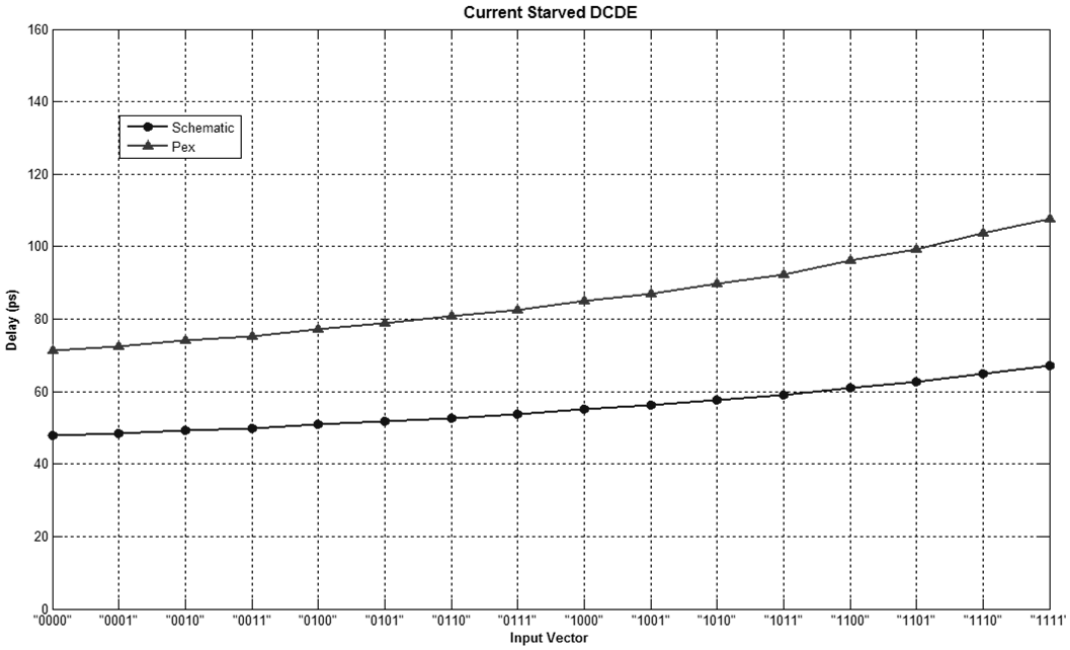

Figure 6.24. Post layout versus schematic delay and linearity for the current starved medium tune delay element

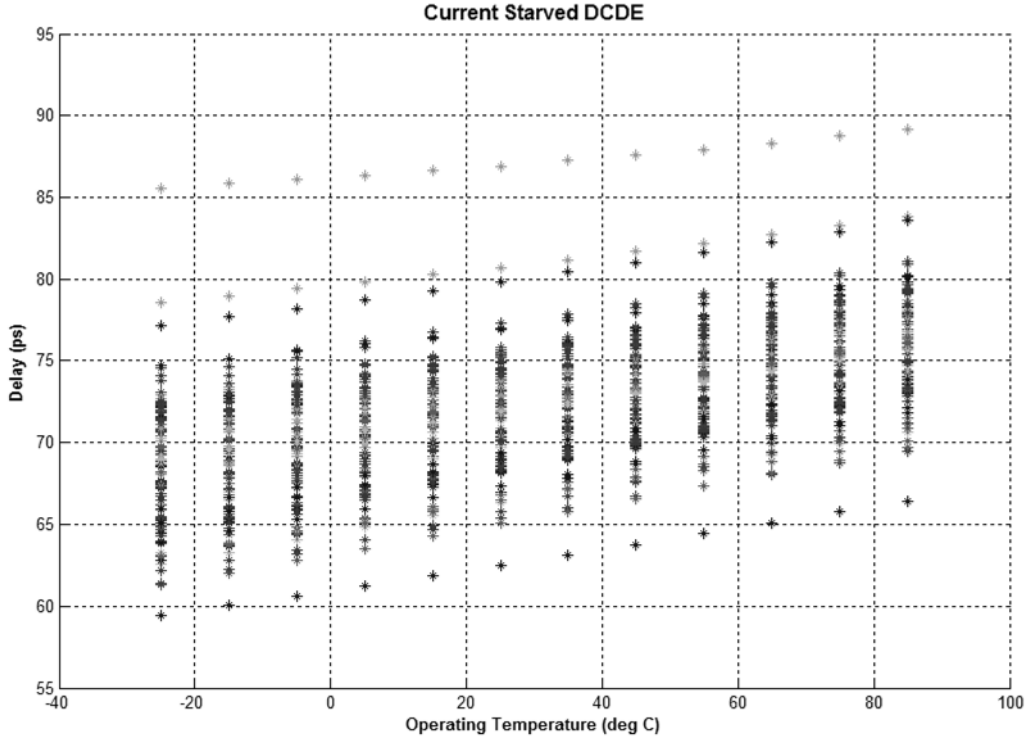

Figure 6.25. Post layout Monte Carlo simulations for the current starved medium tune delay element

The current starved medium tune delay element exhibits a very low temperature dependency and a good probability distribution as seen in figure 6.25 and 6.26 respectively.

Actually it has the lowest temperature dependency and the best probability distribution of all the medium tune elements with 78% of the simulations within 1σ.

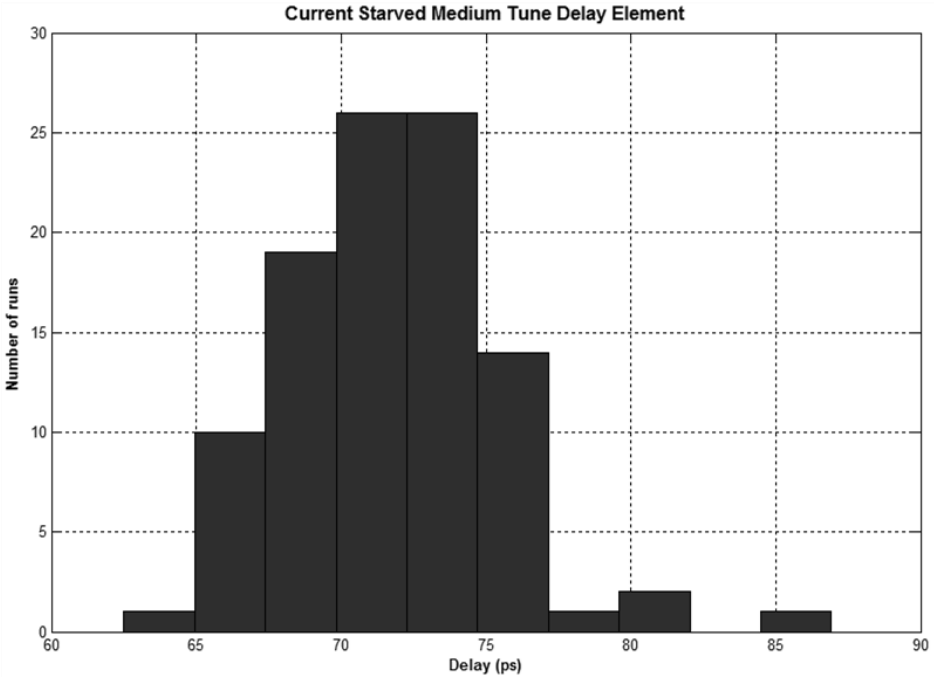

Figure 6.26. Post layout probability distribution for the current starved medium tune delay element

The power consumption is very high compared to the others and is displayed in figure 6.27. Here the dynamic power consumed when the two inverters are switching is clearly visible by the two main peaks on the plot.

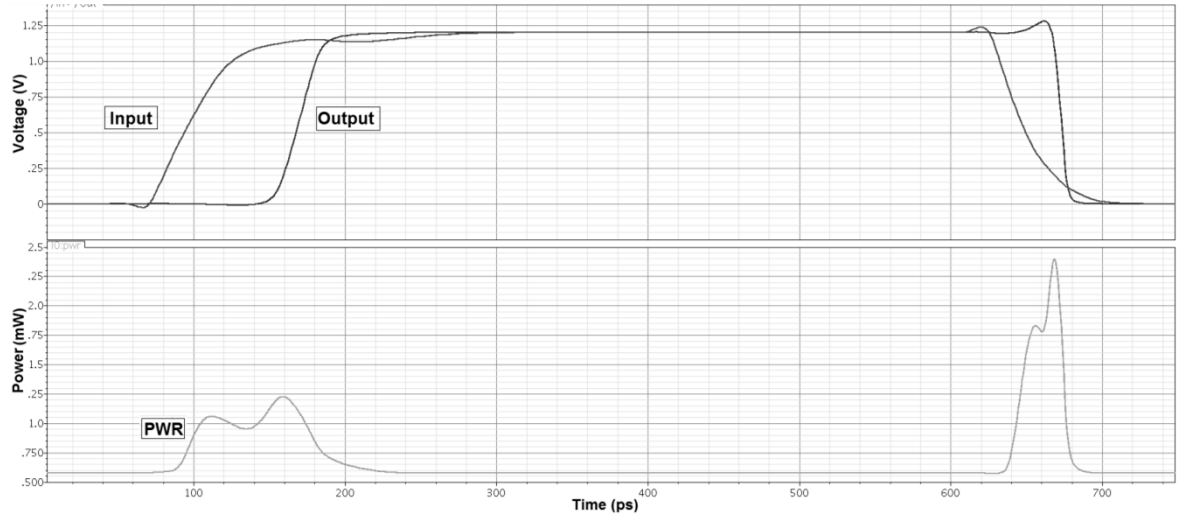

Figure 6.27. Post layout power consumption for the current starved medium tune delay element

When it comes to cycle-to-cycle jitter the current starved medium tune performs the worst of the three medium tune delay elements with $0.206ps\ rms$. This was quite expected since it follows the same trend as the coarse tune delay elements meaning that the architecture is not ideal with respect to jitter noise.

6.5 Shunt Capacitor Medium Tune Delay Element

Figure 6.28 and 6.29 are post layout simulations of the tunability of the shunt capacitor medium tune delay element with all 16 input vector configurations. The linearity is clearly the better of the two tunable medium tune delay elements.

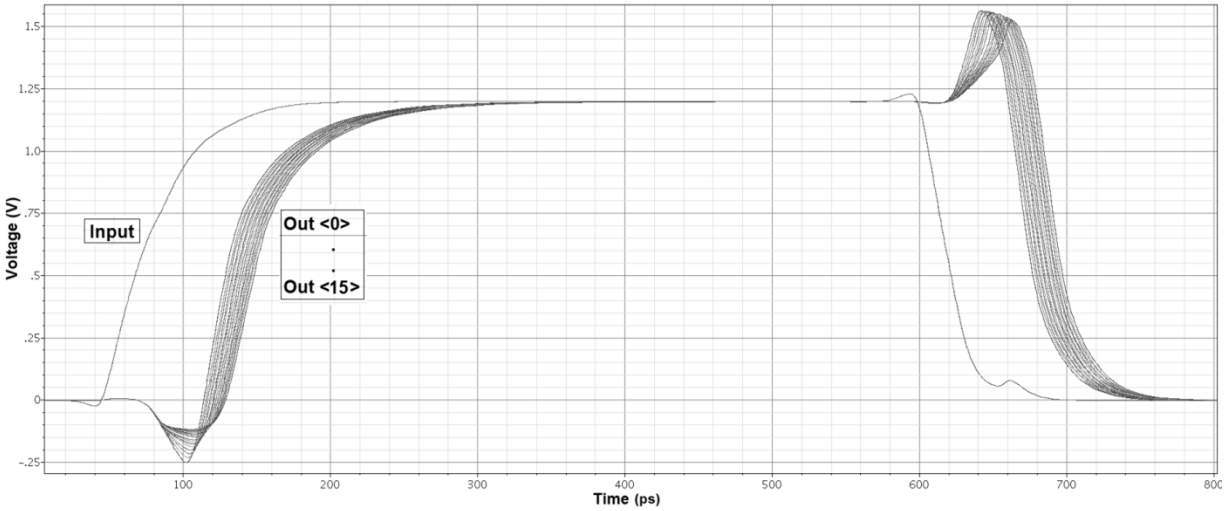

Figure 6.28. Post layout simulation of all input vectors for the shunt capacitor medium tune delay element

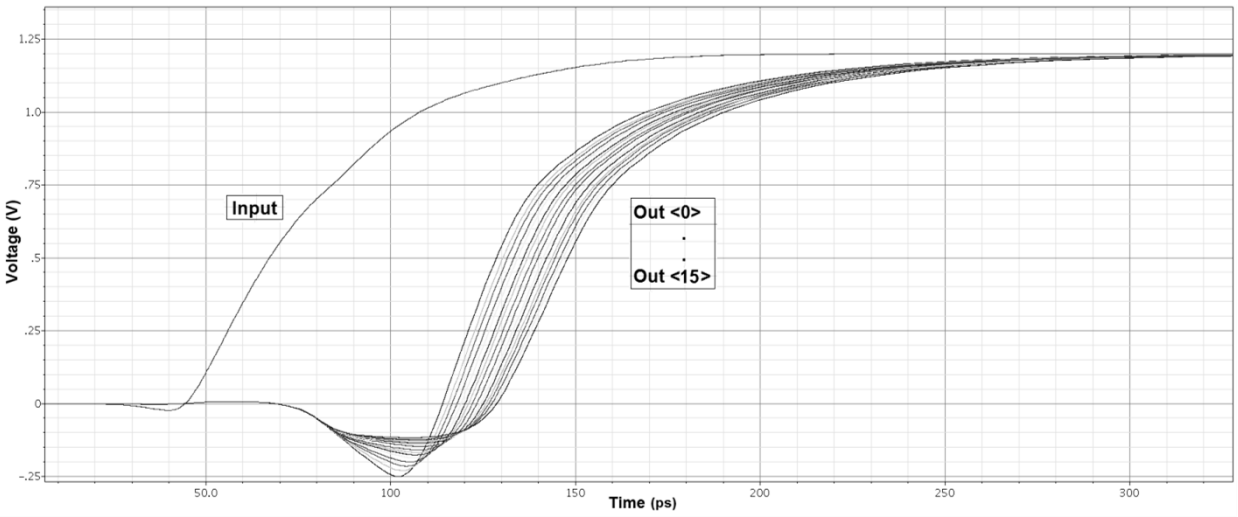

Figure 6.29. Post layout simulation of all input vectors for the shunt capacitor medium tune delay element 2

In figure 6.30 the linearity of the delay element as well as post layout versus schematic delay is depicted. Note that the post layout values affect both the linearity and the delay of the element.

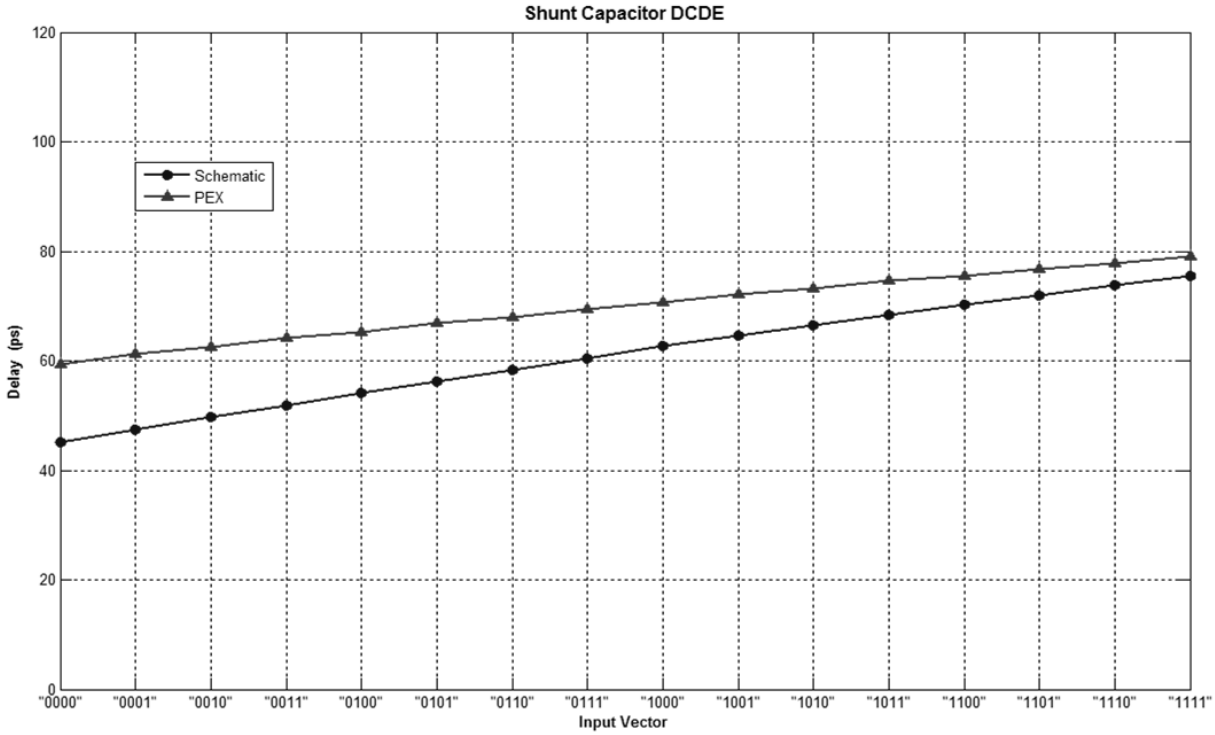

Figure 6.30. Post layout versus schematic delay and linearity for the shunt capacitor medium tune delay element

When it comes to temperature dependency (figure 6.31) it performs the worst of the three medium tune architectures. From figure 6.32 and table 6.1 it can be seen that the shunt capacitor medium tune delay element exhibits moderate process and mismatch-variations compared to the others with 74% of the simulations within 1σ.

Regarding cycle-to-cycle jitter noise performance it exhibits, as the coarse tune shunt capacitor delay element, a moderate amount with $0.102ps\ rms$. This is half of the cycle-to-cycle jitter in the current starved medium tune delay element but twice as much as the benchmark, the non-tunable buffer medium tune. Clearly the results regarding the different test parameters follow the architectures in both the coarse and medium-tune delay elements, giving rise to consistent results.

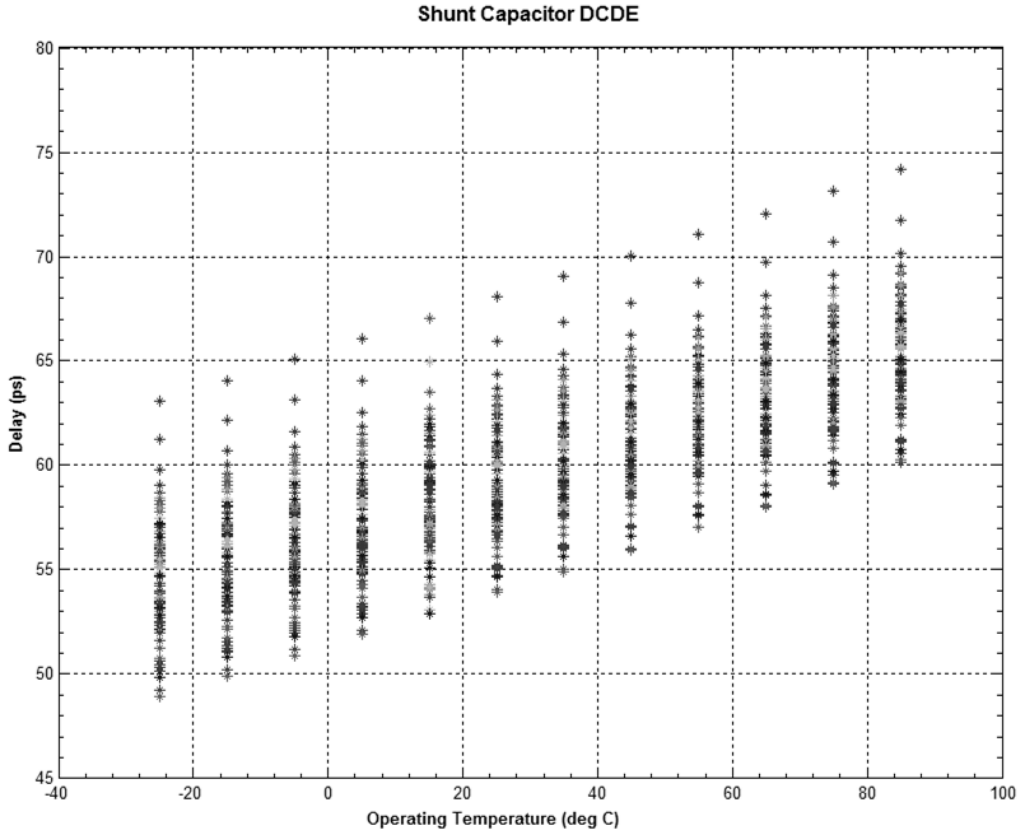

Figure 6.31. Post layout Monte Carlo simulations for the shunt capacitor medium tune delay element

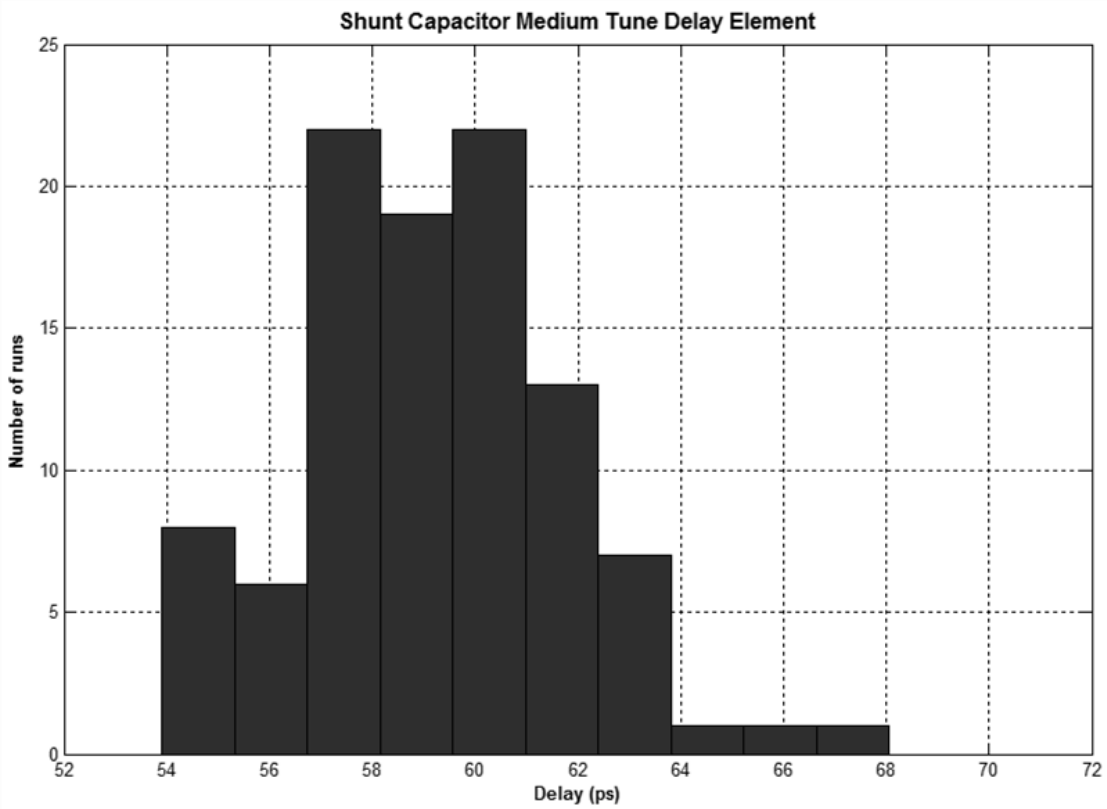

Figure 6.32. Post layout probability distribution for the shunt capacitor medium tune delay element

Power consumption is illustrated in figure 6.33. The dynamic power consumed when the two inverters are switching can be seen clearly as the two peaks in the plot. Compared to the other medium tune delay elements the shunt capacitor has quite low power consumption. It is a lot lower than the current starved medium tune but is slightly higher than the buffer medium tune.

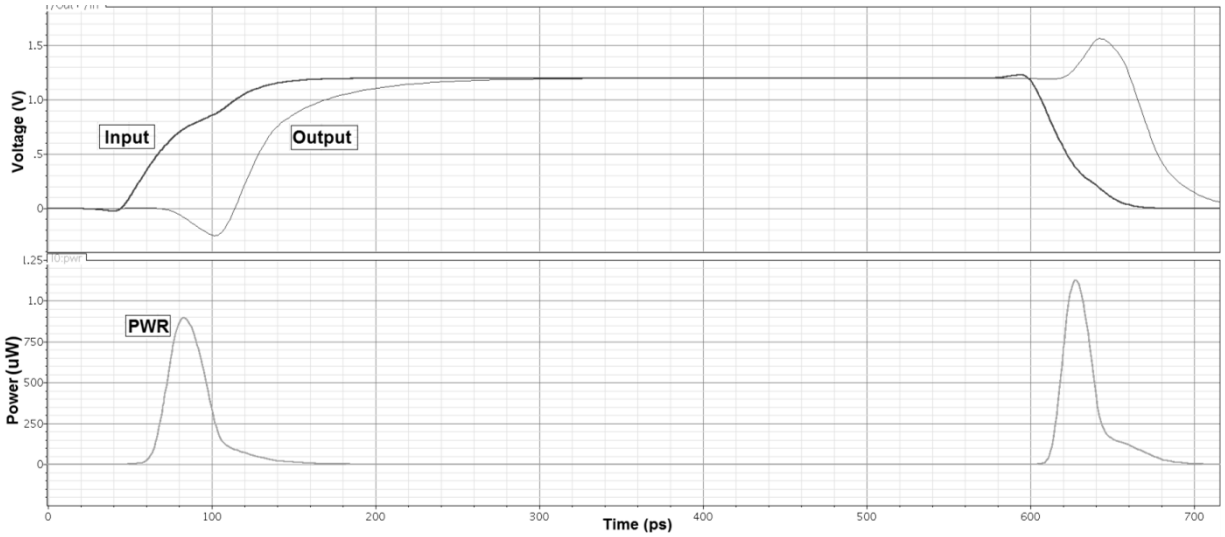

Figure 6.33. Post layout power consumption for the shunt capacitor medium tune delay element

6.6 Buffer Medium Tune Delay Element

The buffer medium tune delay element has no intrinsic tuning but figure 6.34 displays a post layout simulation of the $25ps$ intrinsic delay. It is to be noted, as for all post layout simulations, that the actual delay is approximately 20% higher than the schematic delay due to post layout extracted parasites and routing delay considerations. The actual post layout delay at $27℃$ is $29ps$ compared to the desired schematic $25ps$ delay.

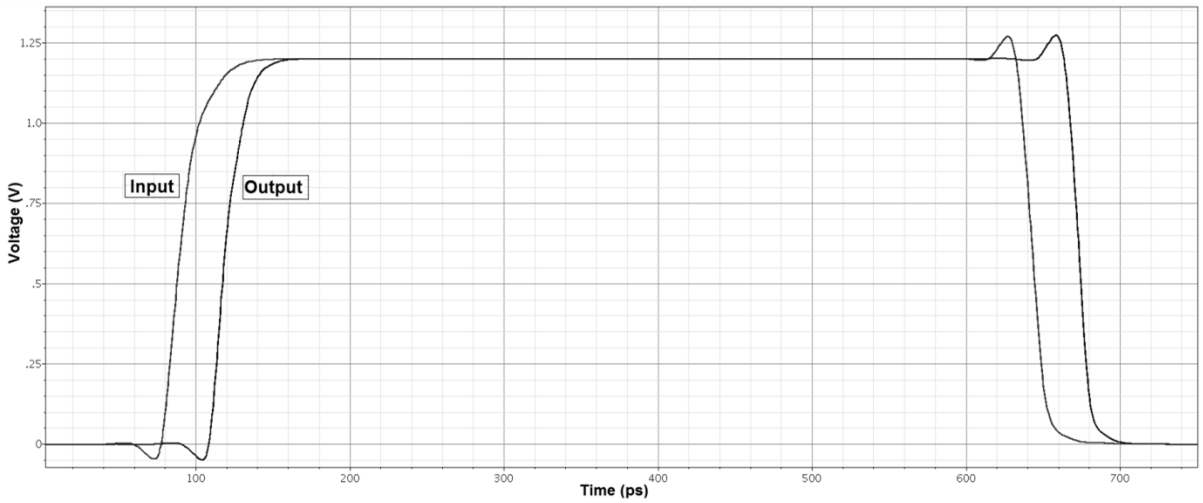

Figure 6.34. Post layout simulation of intrinsic delay for the buffer medium tune delay element

It is clear that the buffer medium tune element is the delay element with the highest resolution, but it lacks the tunability to compensate for the mismatch and temperature variations depicted in figure 6.35 and 6.36. It actually has the highest process and mismatch-variations compared to the others with 72% of the simulations within 1σ. It has the second best temperature dependency, only beaten by the very low temperature dependent current starved medium tune delay element.

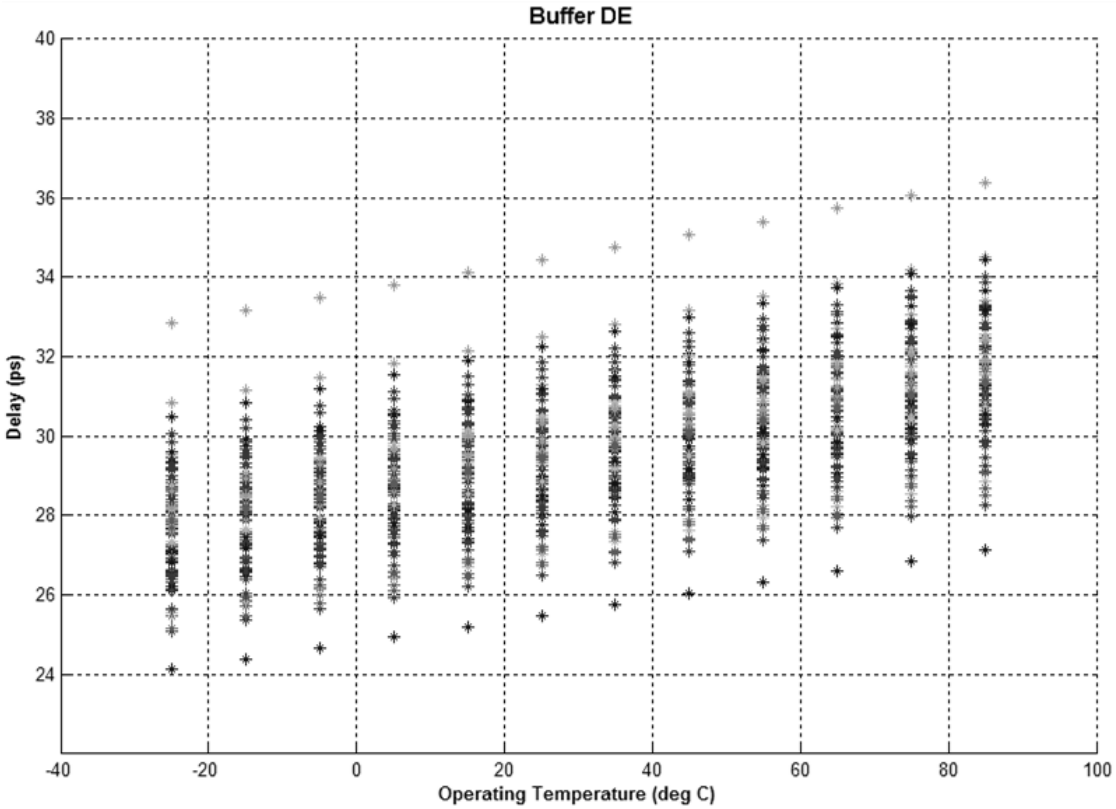

Figure 6.35. Post layout Monte Carlo simulations for the buffer medium tune delay element

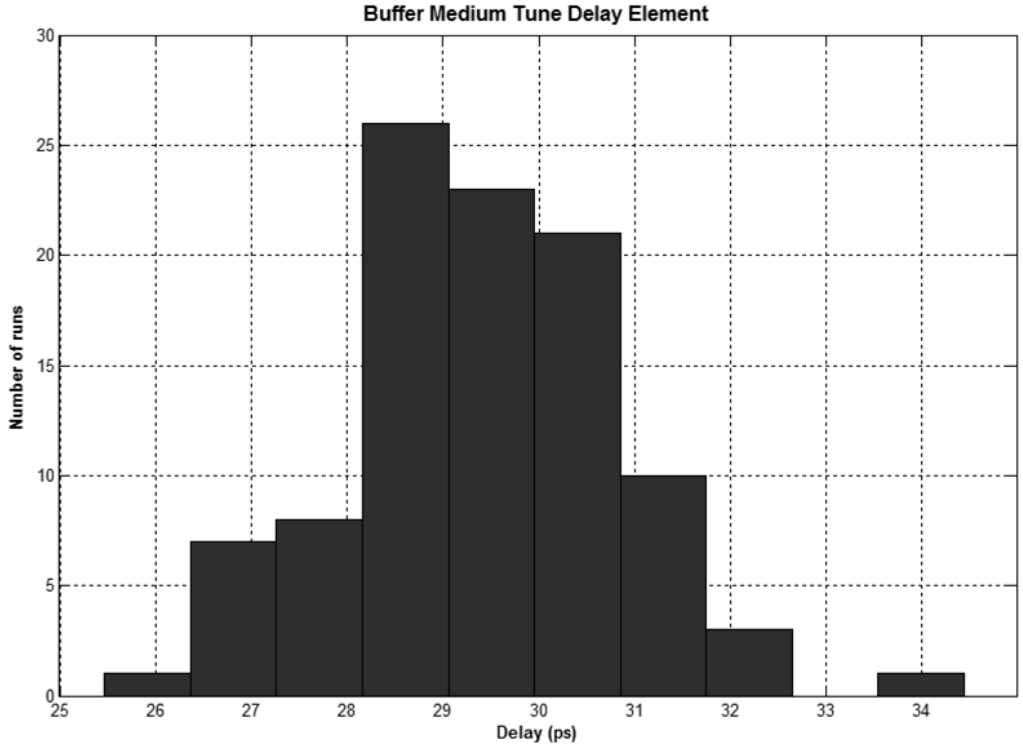

Figure 6.36. Post layout probability distribution for the buffer medium tune delay element

When it comes to power consumption, the buffer medium tune has the lowest power consumption of the architectures. The fast switching results in high resolution and low power due to the short time interval where both the $nMOS$ and $pMOS$ are partially on in the inverters. This reduces the time of the short-circuit current, thus reducing the power consumed. The power consumption can be seen in figure 6.37 and the dynamic power consumed when the inverters are switching can be spotted as the two distinctive peaks on the plot.

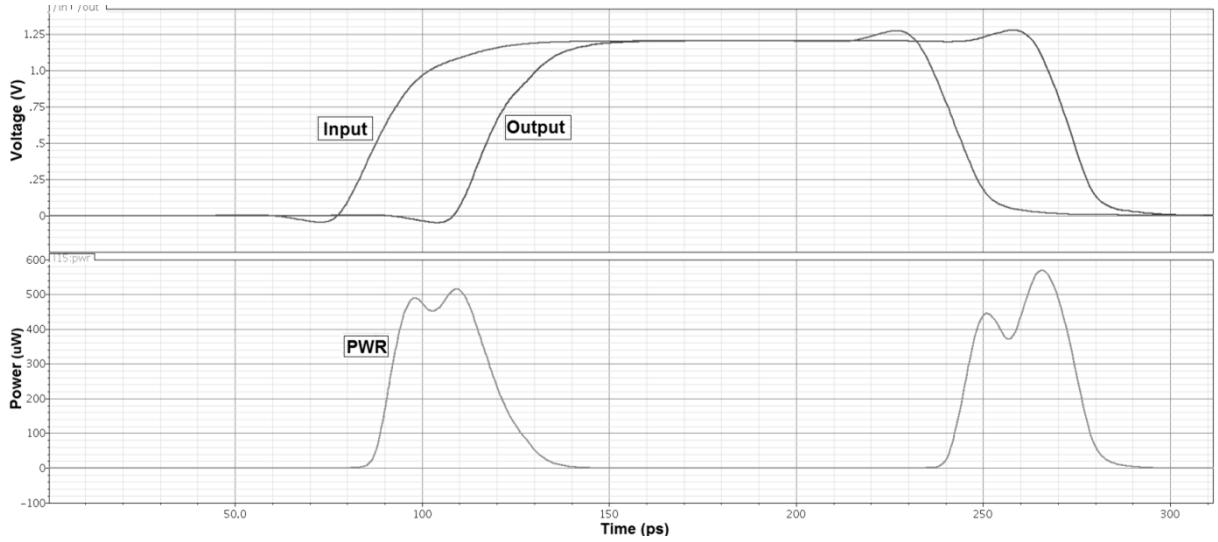

Figure 6.37. Post layout power consumption for the buffer medium tune delay element

Regarding cycle-to-cycle jitter the buffers prove once again to be the benchmark with as little as $0.042ps\ rms$. To put that in perspective this equates to a distance of $1.26 \cdot 10^{-5}m$ in the time domain at the speed of light if used in a radar system. Clearly the non-tunable buffers exhibit the best jitter performance.

6.7 Current Starved Delay Line

A transient simulation of the delay and power consumption of the current starved delay line is shown in figure 6.38 and 6.39 respectively. Figure 6.38 illustrates the fastest setting, meaning that the pulse passes through one CT delay element and one MT delay element before reaching the output. In the fast setting the pulse width reduction is almost not present but it is very distinctive at the slow setting where the pulse width is reduced to around 10% relative to the input PRF. The reason for this is due to the fact that the slowest setting starves the delay elements by intrinsic tuning. In addition the pulse passes through all the delay elements in the delay line and since the current starved delay elements are not symmetric this results in a high degree of pulse width reduction on the output.

It is to be noted that the actual delay is $1.88ns$, not the ideal $1.05ns$. This is due to the added delay through the large multiplexers. In a post layout simulation, routing delays would also affect the total delay.

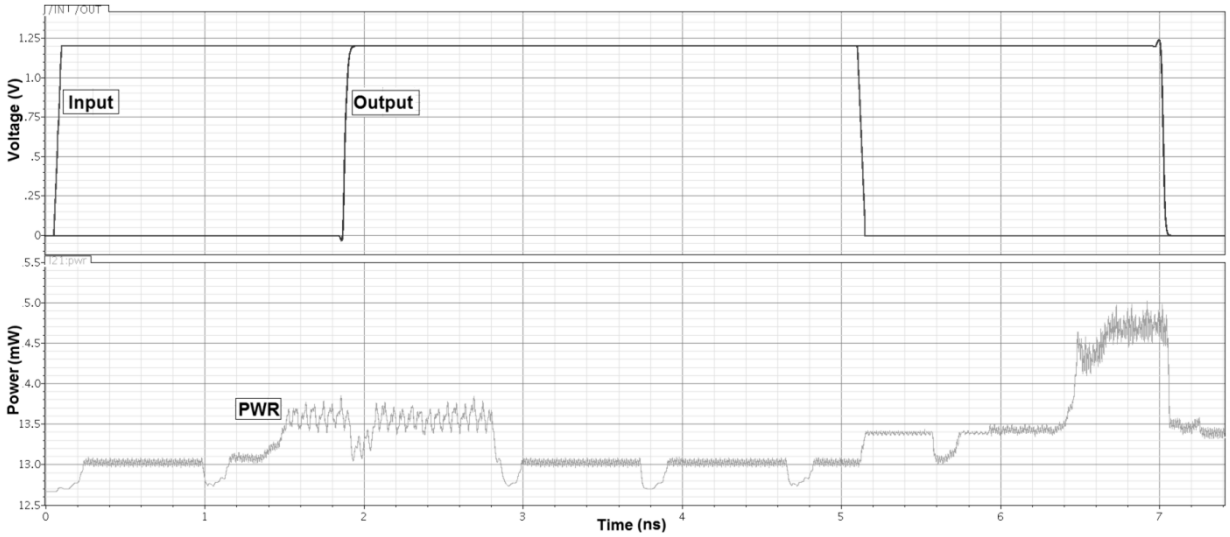

Figure 6.38. Transient simulation displaying delay and power consumption of delay line 1 at the fastest setting

The power consumption is clearly the highest of the three, especially at the fast setting where all of the current sources are turned on. This matches the simulation results from the current starved delay elements.

Even though the current starved delay line suffers from pulse width reduction, it has the lowest temperature dependency of all the delay lines with only a 12.86% delay variation on the slow setting over a temperature range of $110℃$.

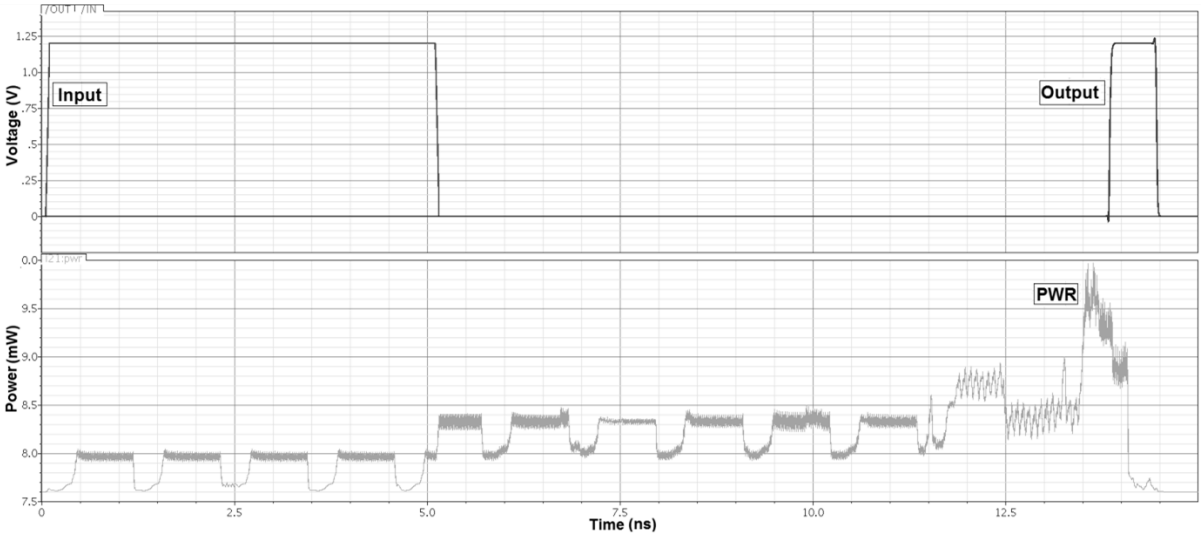

Figure 6.39. Transient simulation displaying delay and power consumption of delay line 1 at the slowest setting

The periodic cycle-to-cycle jitter for the delay line as seen in table 6.2 indicates a value of $1.52ps\ rms$. Since the delay line at the fastest setting only passes through one coarse tune and one medium tune delay element, which the jitter has been measured previously as displayed in table 6.1, the jitter produced by the rest of the circuit (multiplexers) can be estimated. If the noise sources are uncorrelated using eq. 2.5 we get:

$$J_{MUX} = \sqrt{{J_{TOT}}^2 - {J_{DE}}^2} = \sqrt{1.52ps^2 - 1.296ps^2} = 0.793ps$$

where J_{DE} is the summed up jitter from one coarse and one medium tune delay element based on post layout simulations and J_{TOT} is the total delay line jitter.

This is only an estimate but it gives some insight to how much jitter is produced by the multiplexers.

6.8 Shunt Capacitor Delay line

As seen in table 6.2, the pulse width reduction is not present for the shunt capacitor delay line. This can also be seen in figure 6.40 and 6.41 where the delay and power consumption is illustrated.

It is clearly visible that the delay line does not suffer from pulse width reduction, even at the slowest setting. As for the current starved delay line the delay is a larger than the ideal $1.05ns$ due to the added delay through the multiplexers.

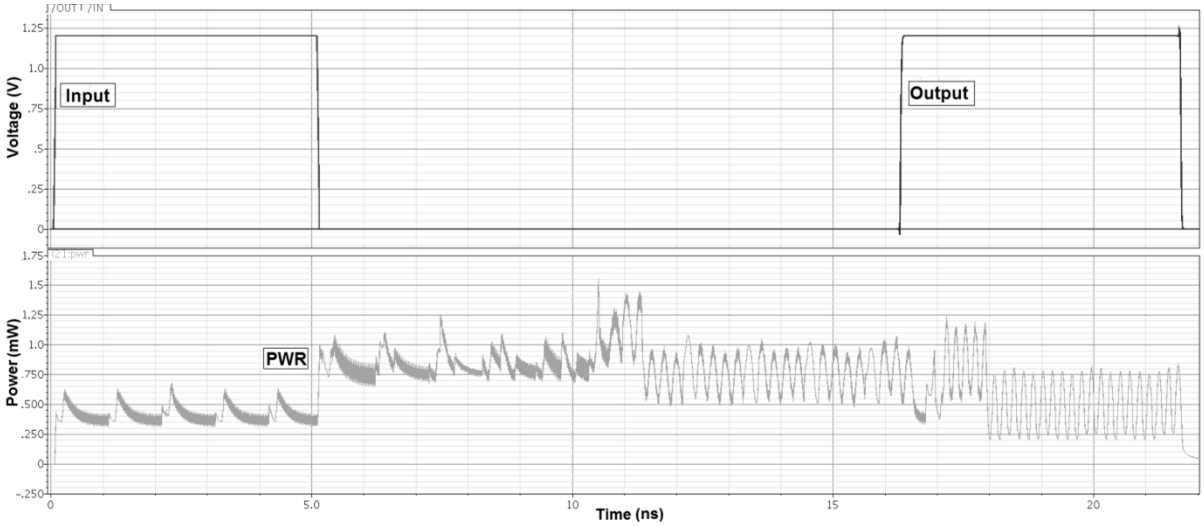

Figure 6.40. Transient simulation displaying delay and power consumption of delay line 2 at the fastest setting

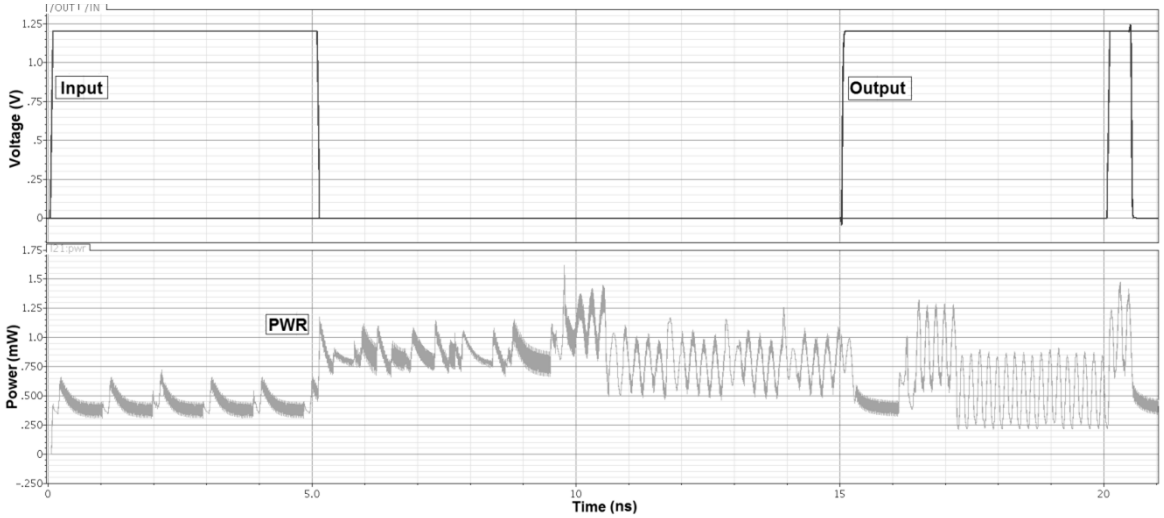

Figure 6.41. Transient simulation displaying delay and power consumption of delay line 2 at the slowest setting

The shunt capacitor delay line has the lowest average power consumption with only $650\mu W$ which is quite low compared to the others. On the other hand it suffers from high temperature dependency with 20.22% delay variation on the fast setting over a temperature range of $110℃$. This is definitely the worst of the group.

Regarding cycle-to-cycle jitter it exhibits quite low values, as expected from the simulated delay elements, with $0.894ps\ rms$. Using equation 2.5 to calculate the estimated multiplexer jitter the result is almost identical to the current starved delay line calculations, indicating consistent results:

$$J_{MUX} = \sqrt{{J_{TOT}}^2 - {J_{DE}}^2} = \sqrt{0.894ps^2 - 0.379ps^2} = 0.809ps$$

where J_{DE} is the summed up jitter from one coarse and one medium tune delay element based on post layout simulations and J_{TOT} is the total delay line jitter.

6.9 Non-Tunable Buffer Delay Line

The non-tunable buffer delay line is a quite good all-rounder. It has moderate power consumption and temperature variations even though it lacks the tunability to compensate for these variations. Figure 6.42 and 6.43 are transient simulations displaying the delay and power consumption at the fast and slow setting respectively. It is noted that the buffer delay line, along with the shunt capacitor delay line, has no pulse width reduction on either setting.

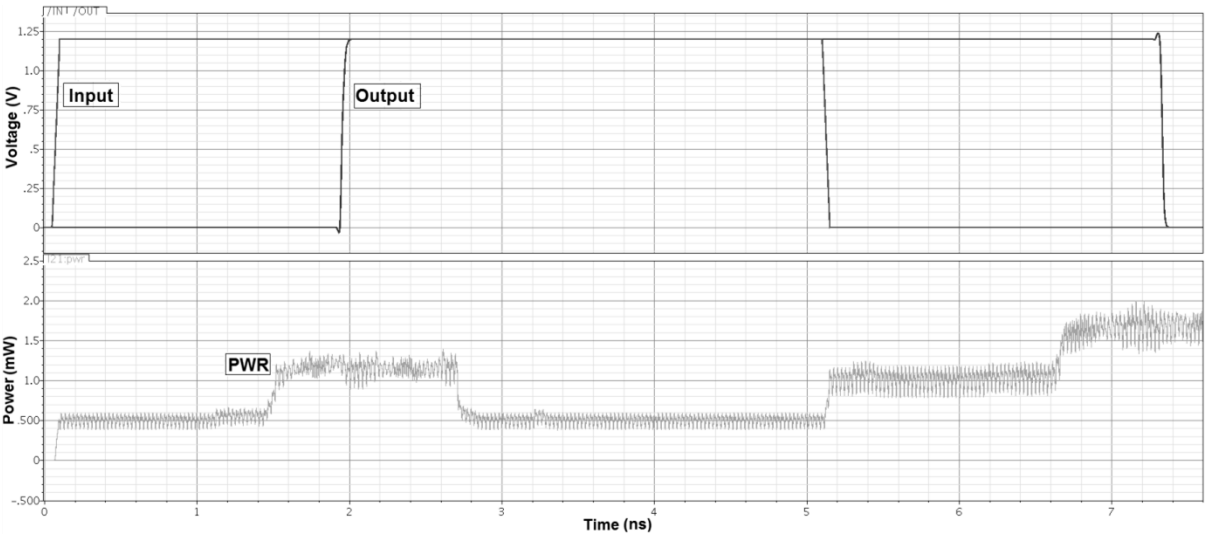

Figure 6.42. Transient simulation displaying delay and power consumption of delay line 3 at the fastest setting

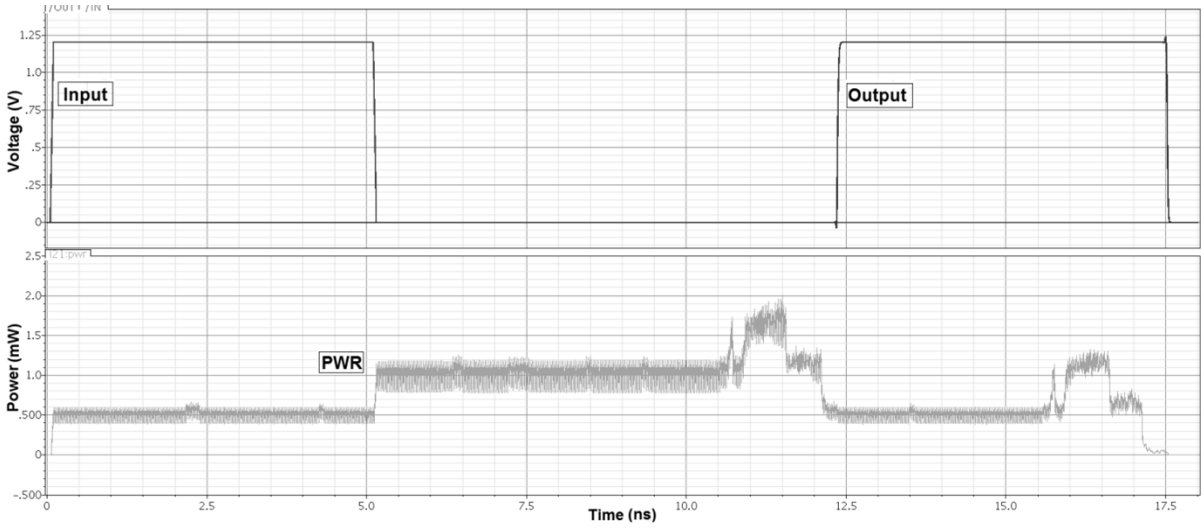

Figure 6.43. Transient simulation displaying delay and power consumption of delay line 3 at the slowest setting

When it comes to the cycle-to-cycle jitter the non-tunable buffer delay line has the lowest jitter noise with $0.743ps\ rms$. This is better compared to the shunt capacitor delay line's $0.894ps\ rms$, but not by the same percentage as the delay element results. This seems a bit odd compared to the results of the delay elements where the buffers had significantly lower cycle-to-cycle jitter noise.

One explanation might be that the non-tunable buffer delay line has more multiplexer jitter even though it does not indicate that using the estimated multiplexer jitter calculations:

$$J_{MUX} = \sqrt{{J_{TOT}}^2 - {J_{DE}}^2} = \sqrt{0.743ps^2 - 0.2881ps^2} = 0.685ps$$

where J_{DE} is the summed up jitter from one coarse and one medium tune delay element based on post layout simulations and J_{TOT} is the total delay line jitter.

The multiplexer jitter is only an estimate and cannot be considered a valid result. The non-tunable buffer delay line uses a much larger 44:1 MUX in the medium tune compared to the 24:1 MUX used in the current starved and shunt capacitor medium tune. It is therefore logical to presume that this is where some of the jitter noise advantage is lost even though it still clearly exhibits the lowest jitter noise, which is an interesting result.

6.10 Self-Compensating DCDE

The main purpose of introducing the digitally controllable delay elements is being able to tune out and compensate for delay variations caused by various sources. To demonstrate this is a Monte Carlo simulation was performed to extract delay variations for the delay elements. In this case the current starved coarse tune element $CT1$ was used. This coarse tune element, as explained in chapter 3, consists of 10 cascaded digitally controllable current starved delay elements. Each of these delay elements has 16 different input vector settings (4 bit) to adjust the delay to the desired value. In this example a uniform delay step of $1.45ns$ for each delay element, which a reasonable value for the current starved coarse tune delay element, is the main goal.

In figure 6.44 the self-compensating ability of the current starved coarse tune delay elements is displayed. The first line (square dots) illustrates the random extracted delay values from the Monte Carlo simulation and it is obviously that the delay is not uniformly distributed. Process and mismatch-variations as well as temperature give each of the 10 delay elements different propagation delays. It is to be noted that the delay elements in the Monte Carlo simulation is set to a medium setting, meaning a input vector of "1000", which gives the most headroom to tune the delay as the delay variations may be positive or negative.

The second line (triangular dots) illustrates the theoretical delay values after adjusting the delay elements and self-compensating for the delay variations based on post layout simulations. Clearly the tuned values represent an almost completely uniform distribution.

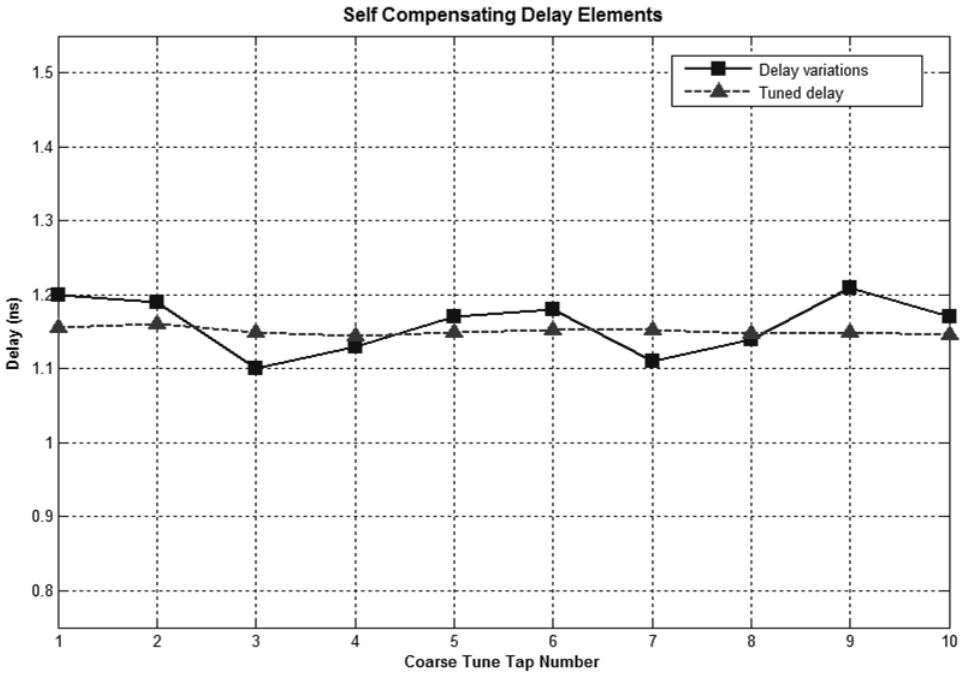

Figure 6.44. Self-compensating ability of the current starved coarse tune delay elements by individual tuning

In table 6.4 the simulated delay deviation from the desired value and the tuning of the input vectors of the delay elements to achieve a uniform delay distribution is shown. This is where the tunable delay elements really come to their use. The static process and mismatch-variations and the temperature variations are almost completely compensated for. The non-tunable buffers could not achieve this individual self-compensating, even though it can compensate somewhat by changing the multiplexer settings to the medium tune by steps of $25ps$. A mix of the coarse and medium tune delay elements is also possible in the tunable delay lines to achieve even higher resolution of the tuning.

Delay Element	***Original Input Vector***	***MC Delay*** ns	***Desired Delay*** ns	***Diff*** ps	***Tuned Input Vector***	***Resulting Delay***
1	"**1000**"	1.50	1.45	-50	"**0100**"	1.455
2	"**1000**"	1.49	1.45	-40	"**0101**"	1.454
3	"**1000**"	1.40	1.45	50	"**1011**"	1.448
4	"**1000**"	1.43	1.45	20	"**1001**"	1.444
5	"**1000**"	1.47	1.45	-20	"**0111**"	1.455
6	"**1000**"	1.48	1.45	-30	"**0110**"	1.454
7	"**1000**"	1.41	1.45	40	"**1010**"	1.448
8	"**1000**"	1.44	1.45	10	"**1001**"	1.454
9	"**1000**"	1.51	1.45	-60	"**0010**"	1.449
10	"**1000**"	1.47	1.45	-20	"**0111**"	1.455

Table 6.4. Comparison of BMT with and without back gate tuning circuitry

The tuned input vectors now produce a map of the settings required to achieve a uniformly distributed delay for these specific variations.

To implement this is in real life, on-chip, creating a map of all the input vector settings for all the delay elements within the delay lines requires an automated measurement setup. There are a total of 272 bits of intrinsic tuning in the two tunable delay lines and another 28 bits of multiplexer settings making up a lot of combinations to test out. An accurate measurement of the delay is required which can be achieved using a novel, non-inverting ring oscillator in combination with an accurate clock reference [47]. If the delay can be measured on-chip relative to an accurate external reference this can be performed for all the delay configurations using a microcontroller and be stored for later use. If the delay for all delay line configurations were measured at different temperatures as well it would be possible to measure the temperature and looking it up in the map to see which bit configurations were required. This could then be adjusted on-the-fly and constantly produce the desired delay accurately.

6.11 Reduced Temperature Dependency Delay Element

The proposed temperature dependency delay element using back gate tuning/biasing was implemented into an inverter as well as the buffer medium tune delay element. In the inverter only $nMOS$ biasing is implemented as only the falling output edge is of interest. In the buffer medium tune delay element both $nMOS$ and $pMOS$ biasing was implemented to affect both the falling edge of the first inverter and the rising edge of the second inverter. It is to be noted that only adjusting the positive edge of the signal will result in a small degree of asymmetry.

The effect of the back gate biasing for the inverter and BMT is shown in table 6.5 and 6.6 respectively. Temperature dependency has been reduced and the intrinsic delay is considerably lower. The inverter has 12% lower intrinsic delay and 4.5% lower temperature dependency whereas the BMT has almost 9% lower intrinsic delay and 3% less temperature dependency.

Temperature (°C)	**Delay** (ps)	
	Inverter	**Inverter with $nMOS$ back gate biasing**
$\mathbf{-25}$	9.85	8.67 @ $V_{BSn} = 436mV$
$\mathbf{85}$	11.41	9.66 @ $V_{BSn} = 526mV$
$\boldsymbol{Diff}$ (%)	15.84%	11.42%

Table 6.5. Comparison of inverter with and without $nMOS$ back gate tuning circuitry

It is to be noted that the simulations are based on schematics not post layout. The effect of the back gate biasing can probably be fine-tuned or redesigned with other design architectures to achieve even better performance but the main principle and possibility of reducing temperature dependency as well as simultaneously reducing the intrinsic delay is demonstrated.

Temperature (℃)	**Delay** (ps)	
	$\boldsymbol{BMT}$	$\boldsymbol{BMT}$ **with** $\boldsymbol{nMOS}$ **and** $\boldsymbol{pMOS}$ **back gate biasing**
$\mathbf{-25}$	22.98	21.02 @ $V_{BSn} = 436mV$ $V_{BSp} = -328mV$
85	26.50	23.60 @ $V_{BSn} = 525mV$ $V_{BSp} = -383mV$
$\boldsymbol{Diff}$ (%)	15.32%	12.27%

Table 6.6. Comparison of BMT with and without back gate tuning circuitry

7 Conclusion

In this thesis 2 different types of digitally programmable delay elements with intrinsic tunability were proposed to compensate for the increased influence of environmental and mismatch-variations in delay lines due to high resolution and small transistor sizing. In addition to being tunable the proposed digitally programmable delay element architectures were designed to exhibit low jitter noise by increasing the slew rate. The basis of comparison was the non-tunable buffer delay element which is considered to be the benchmark of jitter noise performance. The delay element architectures were also compared with respect to other important parameters such as; power consumption, temperature dependency, process and mismatch-variations, INL, DNL, resolution, and tunability.

It was shown that the proposed digitally programmable delay element architectures introduced different advantages and disadvantages, whereas the common main advantage is the self-compensating ability to tune out delay variations. These tuned input vector settings could be saved and used constructively to achieve a uniform delay under varying conditions.

The digitally programmable current starved delay element architecture, in addition to being self-compensating, proved to exhibit very low temperature dependency. The tradeoffs for this architecture were the high jitter noise, high power consumption and pulse width reduction due to asymmetry which limits the maximum PRF and number of cascaded delay elements. Non-linearity was also present but was not a big issue even though it slightly affected the tunability.

The digitally programmable shunt capacitor delay element architecture, in addition to being self-compensating, proved to have a moderate amount of jitter noise and low INL and DNL values. The main drawbacks for this architecture were the large consumed Silicon area and the high temperature dependency.

It is proved that the non-tunable buffers used as a basis of comparison was the benchmark of jitter noise performance even though it appeared to lose some of its jitter advantage due to the larger multiplexers in the delay line. The main drawback was obviously not being self-compensating.

A technique of reducing the temperature dependency of a delay element further, while simultaneously reducing the intrinsic delay by the use of back gate tuning, was also presented.

Implementing each of the different delay element architectures into a tapped delay line provided consistent and similar results compared to the delay element architectures themselves. The self-compensating advantage of the digitally programmable delay elements can be utilized in high precision applications.

7.1 Further research

The results presented in this thesis give rise to some interesting possibilities for further research, such as:

- Implement the proposed back gate tuned buffer delay element in a delay line to see how it compares to the other delay lines
- Implement back gate tuning in the proposed digitally programmable delay element architectures to see how it compares to the other delay elements
- Symmetric design of the current starved digitally programmable delay element architecture by current starving both the rising and falling edge of the propagated signal.
- Fabricate chip from the designed circuit and perform accurate measurements for all tuning combinations and delay settings. Save all the settings to make a map of all delay configurations to see if it is possible to achieve a uniform delay over a wide range of parameters using the self-compensating abilities of the tunable delay elements.
- Implement of other types of delay element architectures to compare with the proposed current starved and shunt capacitor architectures. A differential architecture may be of interest due to its theoretically reduced noise by common mode rejection.

By exploring some of these possibilities in the future, delay elements with even better properties and accuracy might be achieved to be used in high precision applications.

8 Appendix A: Layout

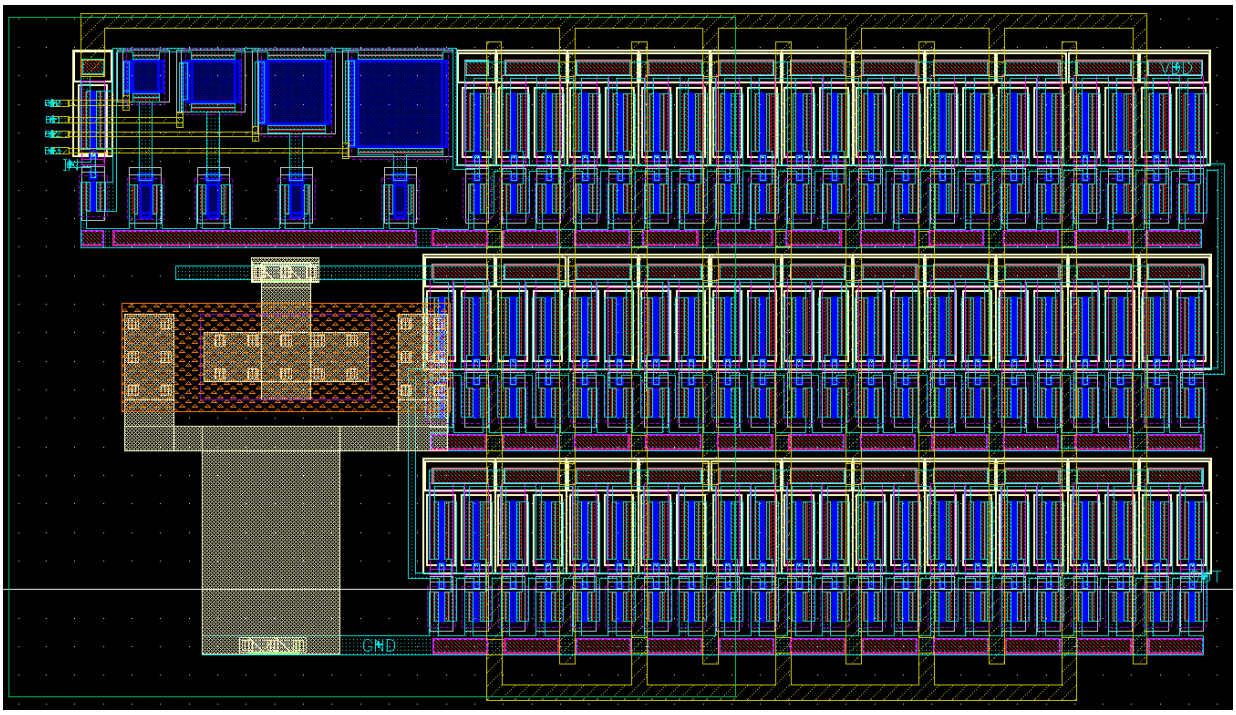

Figure 8.1. Layout of shunt capacitor coarse tune delay element

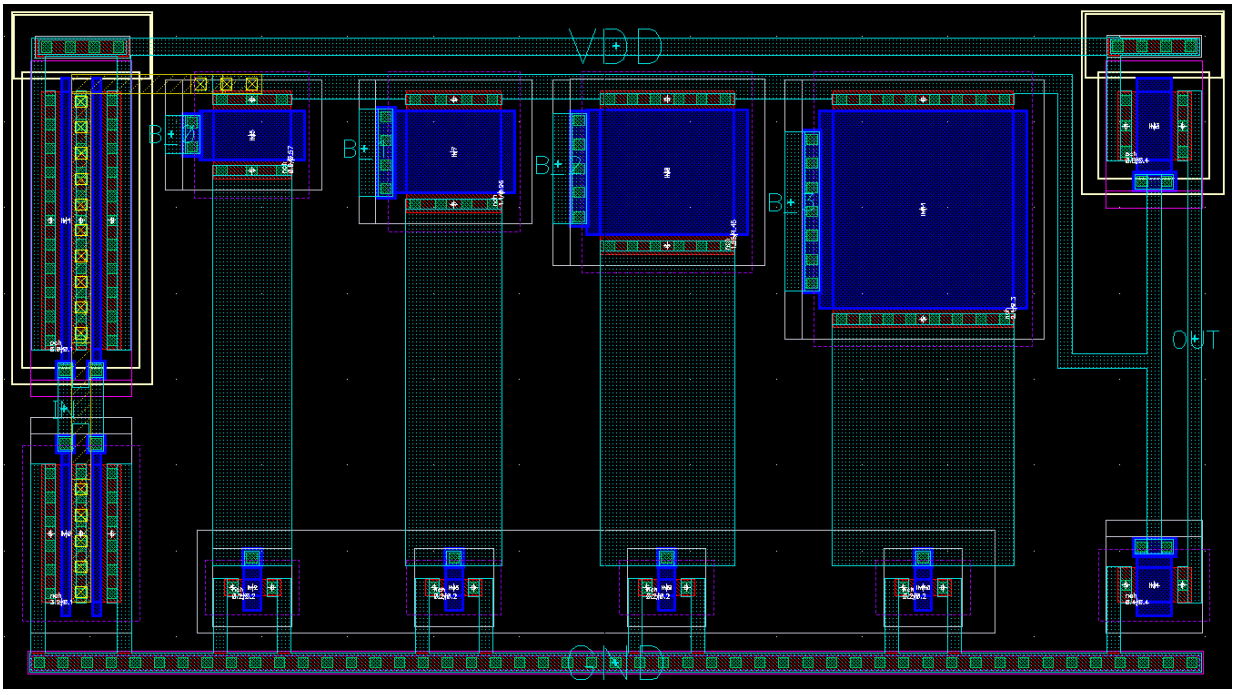

Figure 8.2. Layout of shunt capacitor medium tune delay element

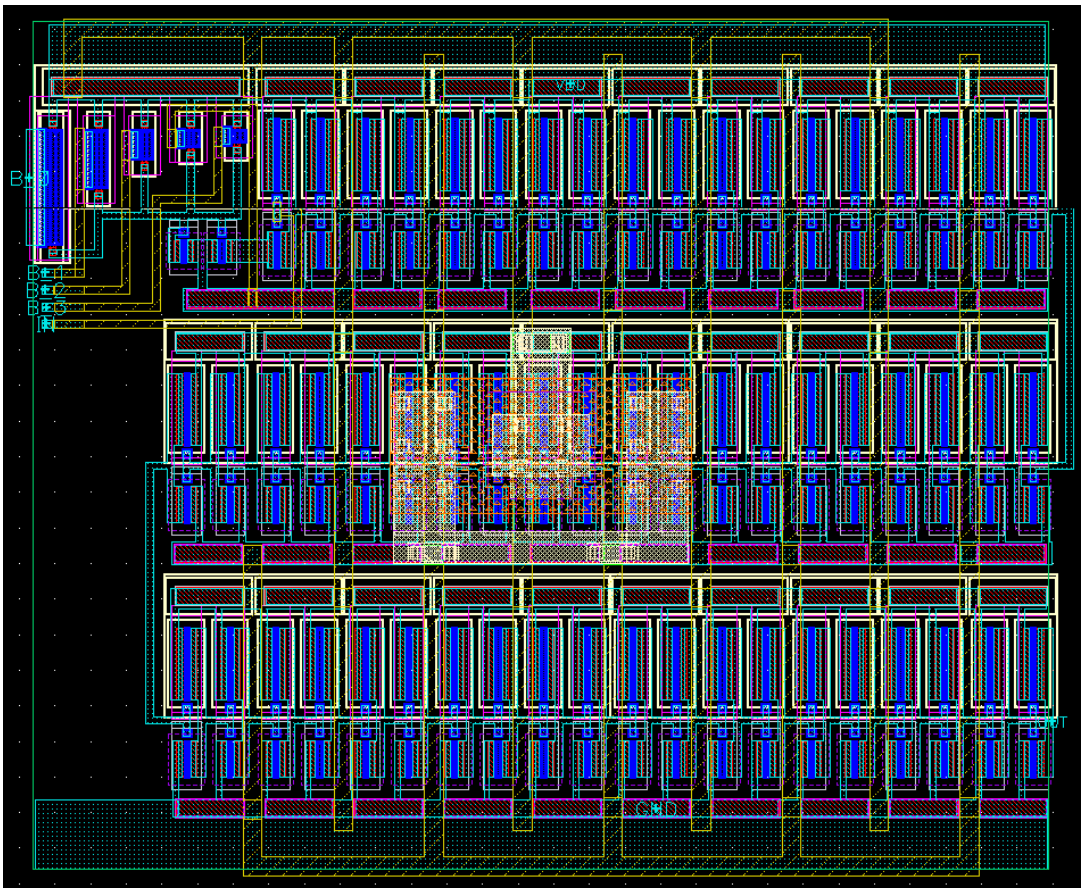

Figure 8.3. Layout of current starved coarse tune delay element

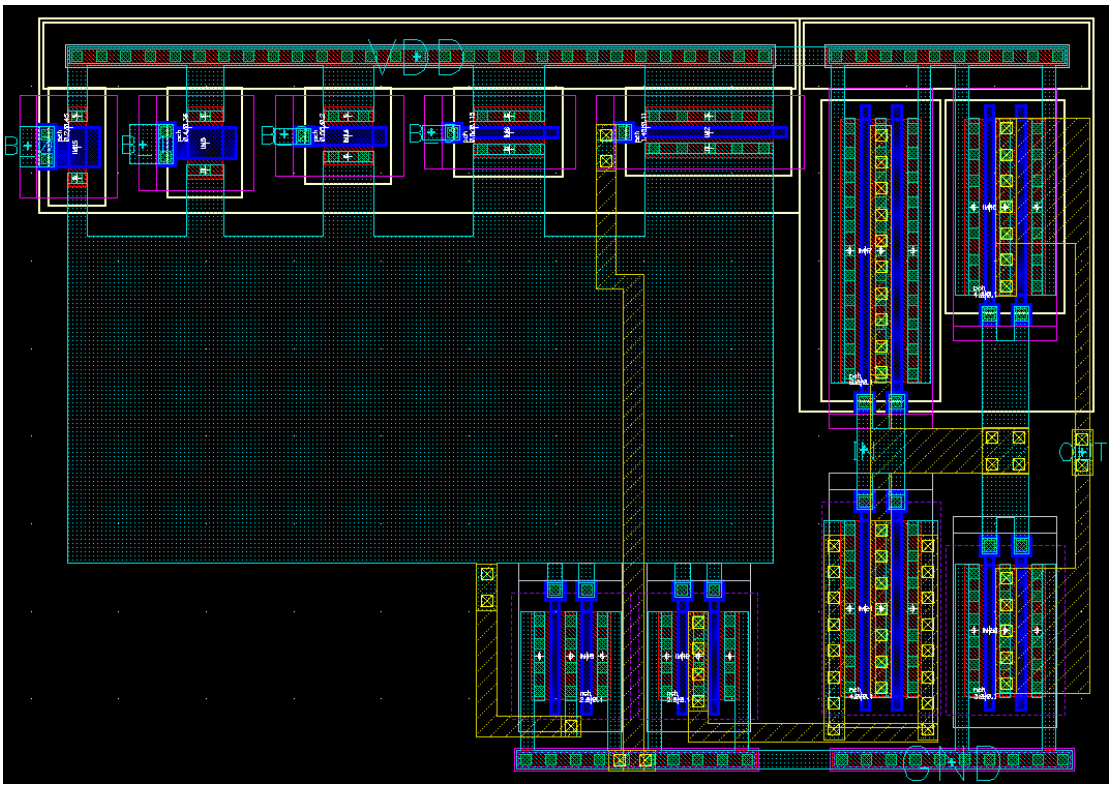

Figure 8.4. Layout of current starved medium tune delay element

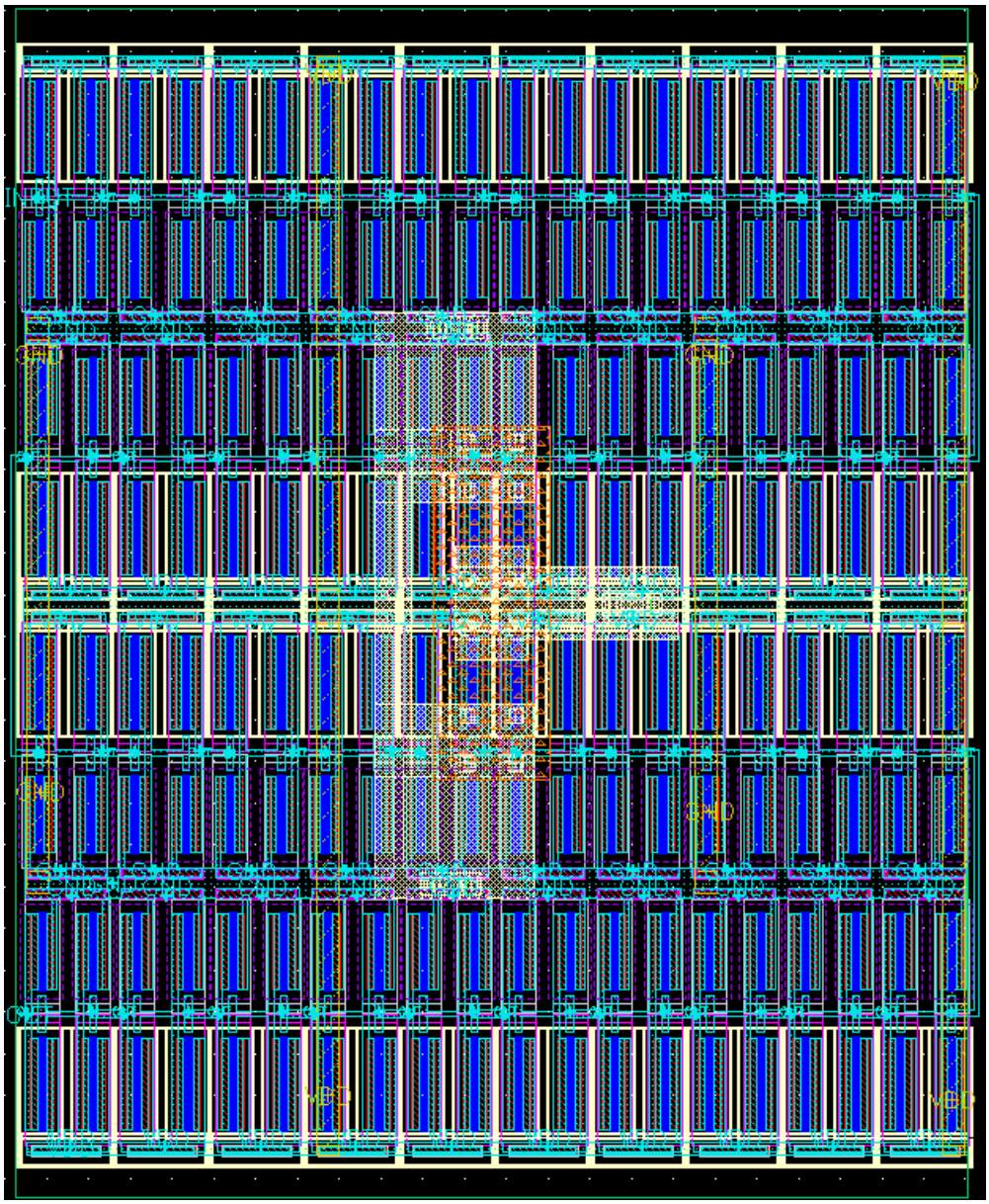

Figure 8.5. Layout of buffer coarse tune delay element

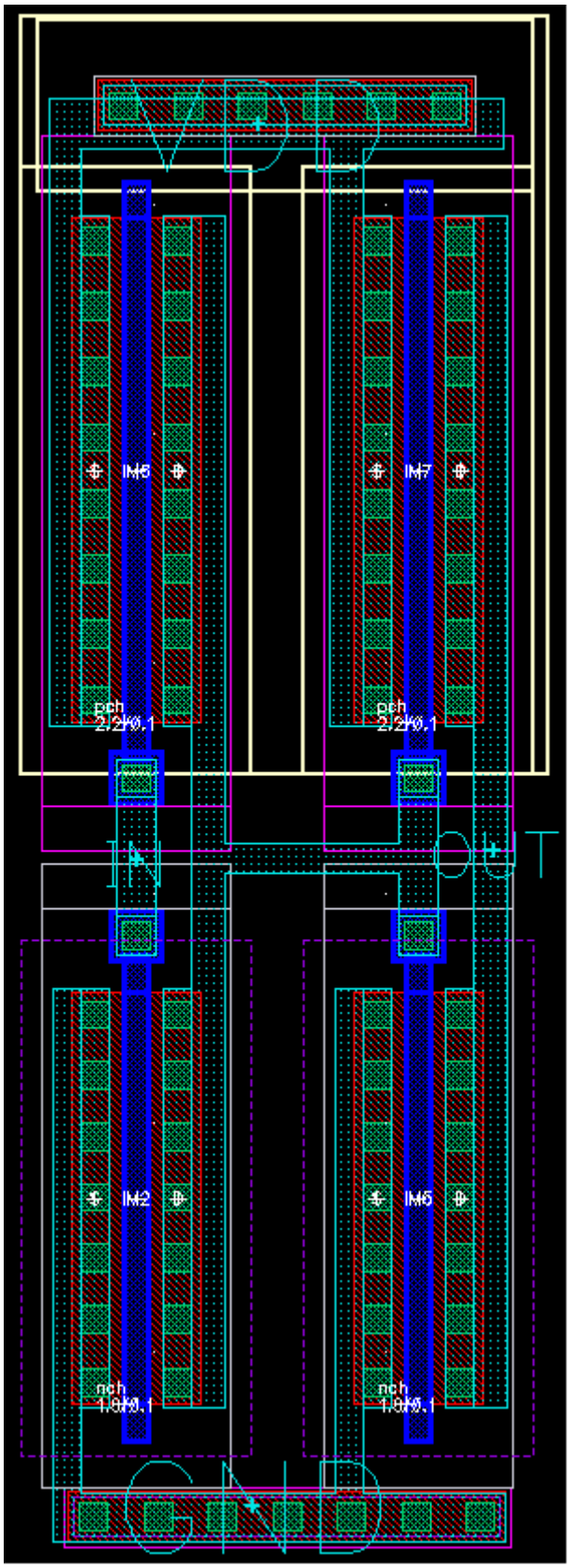

Figure 8.6. Layout of buffer medium tune delay element

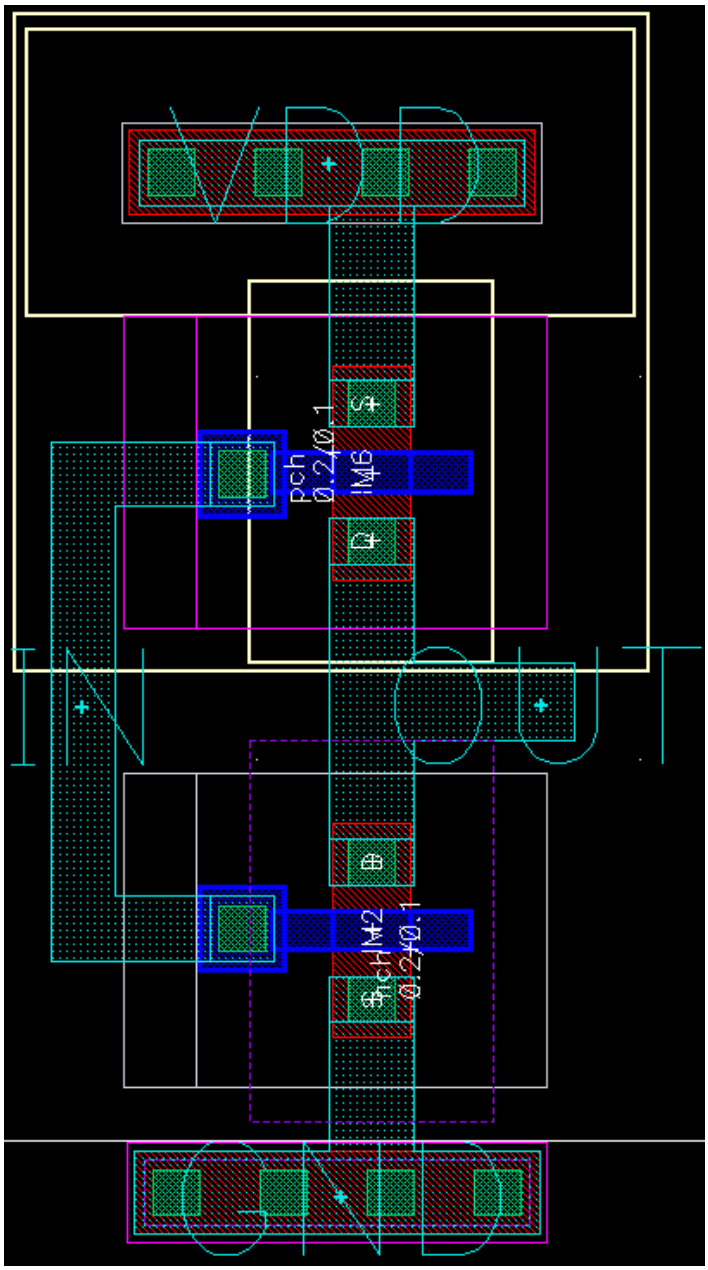

Figure 8.7. Layout of inverter

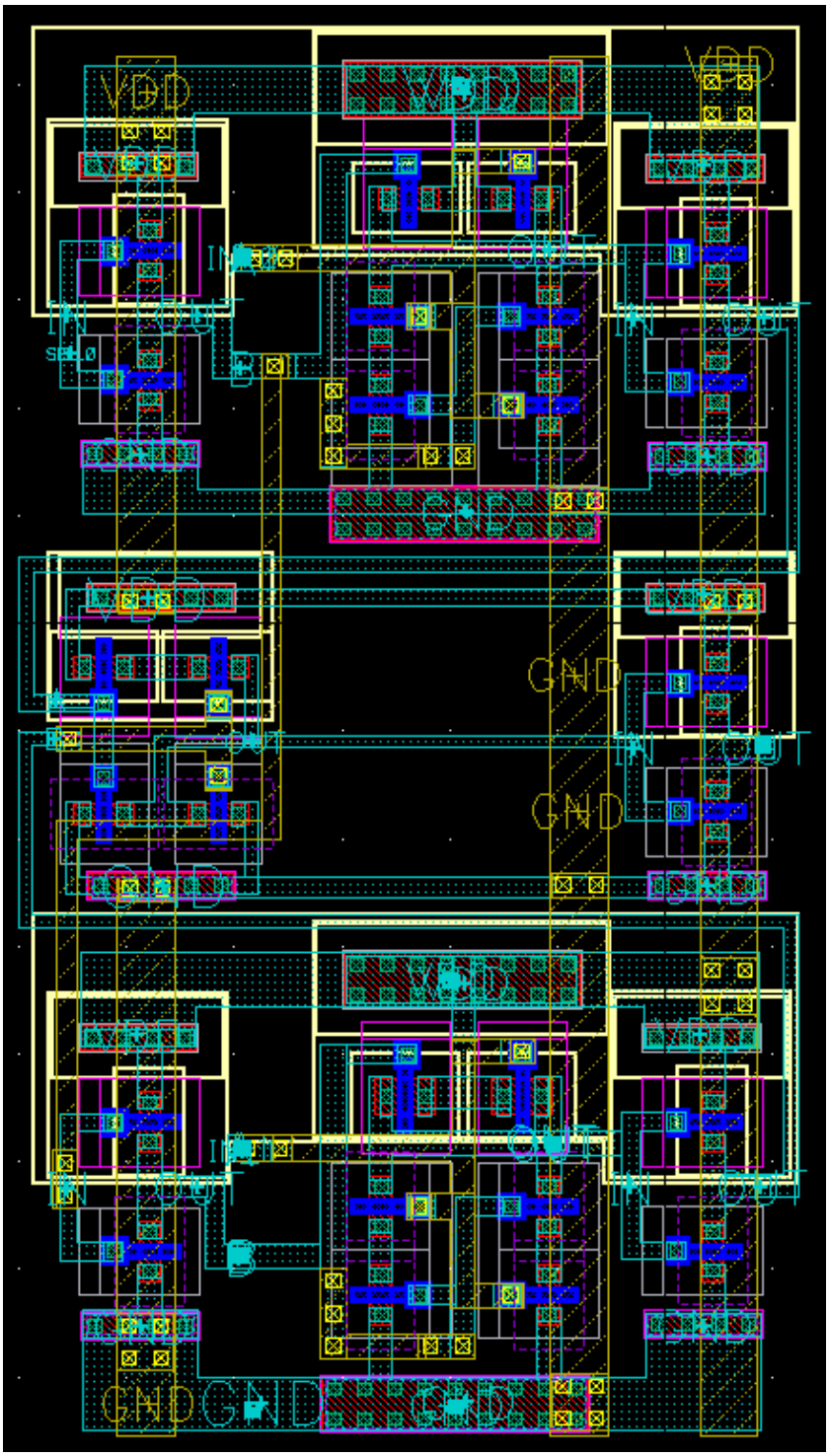

Figure 8.8. Layout of 2:1 MUX

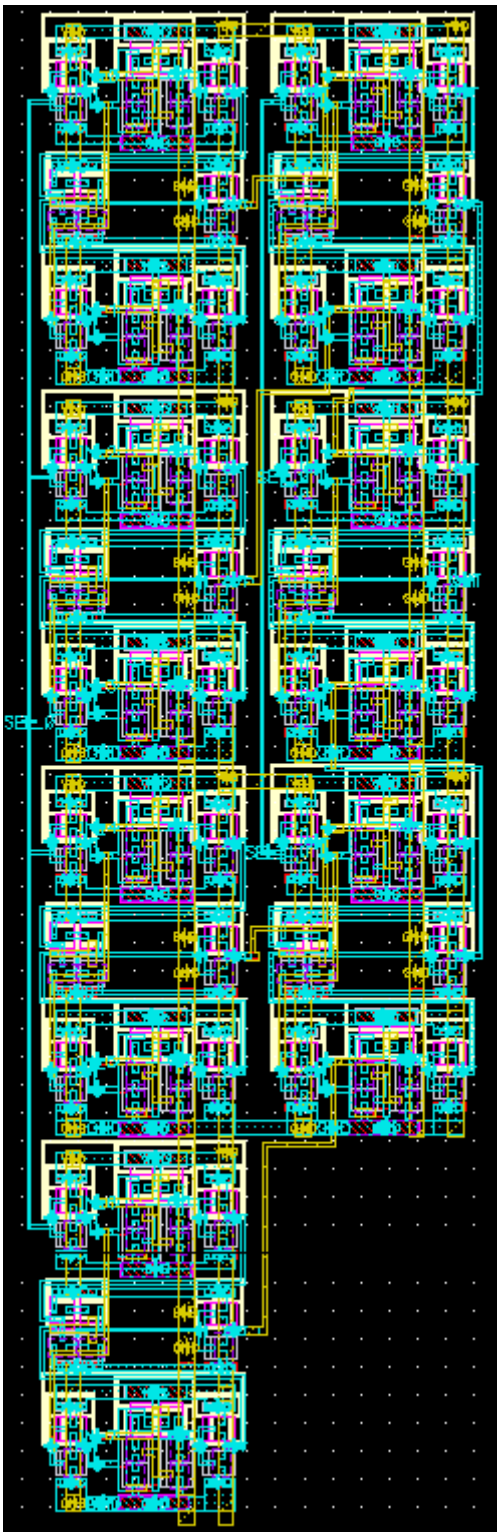

Figure 8.9. Layout of 8:1 MUX

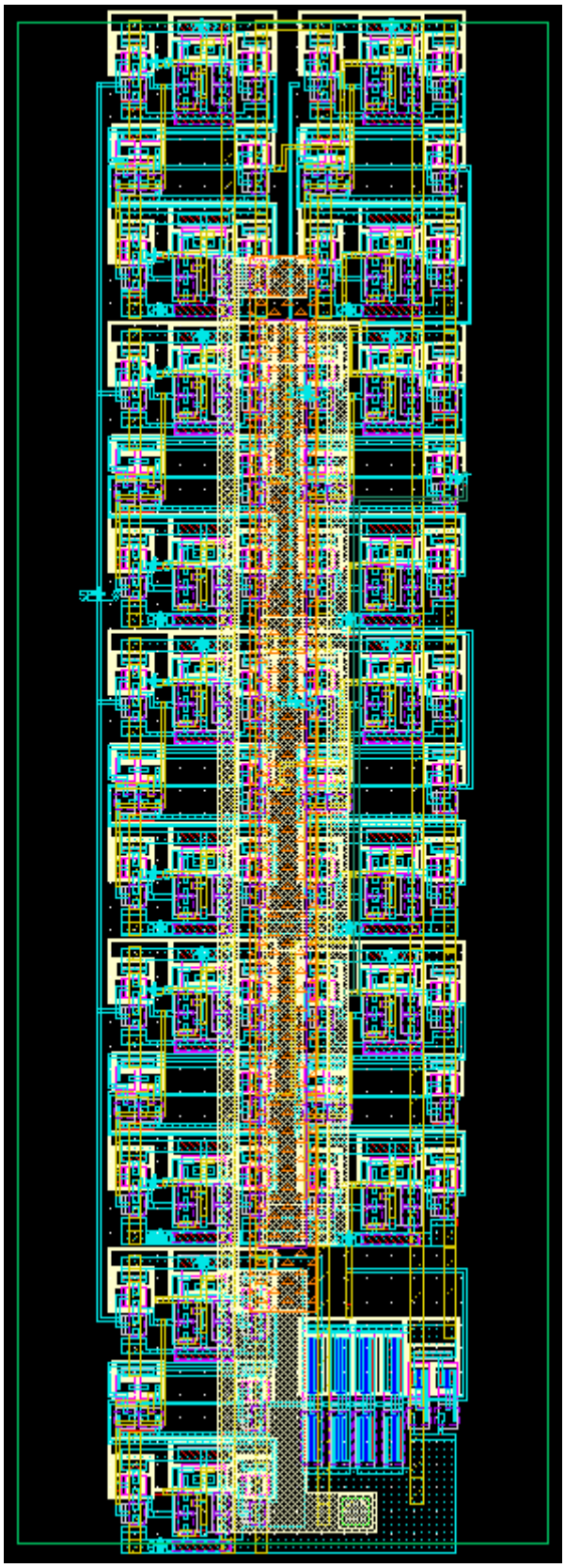

Figure 8.10. Layout of 10:1 MUX

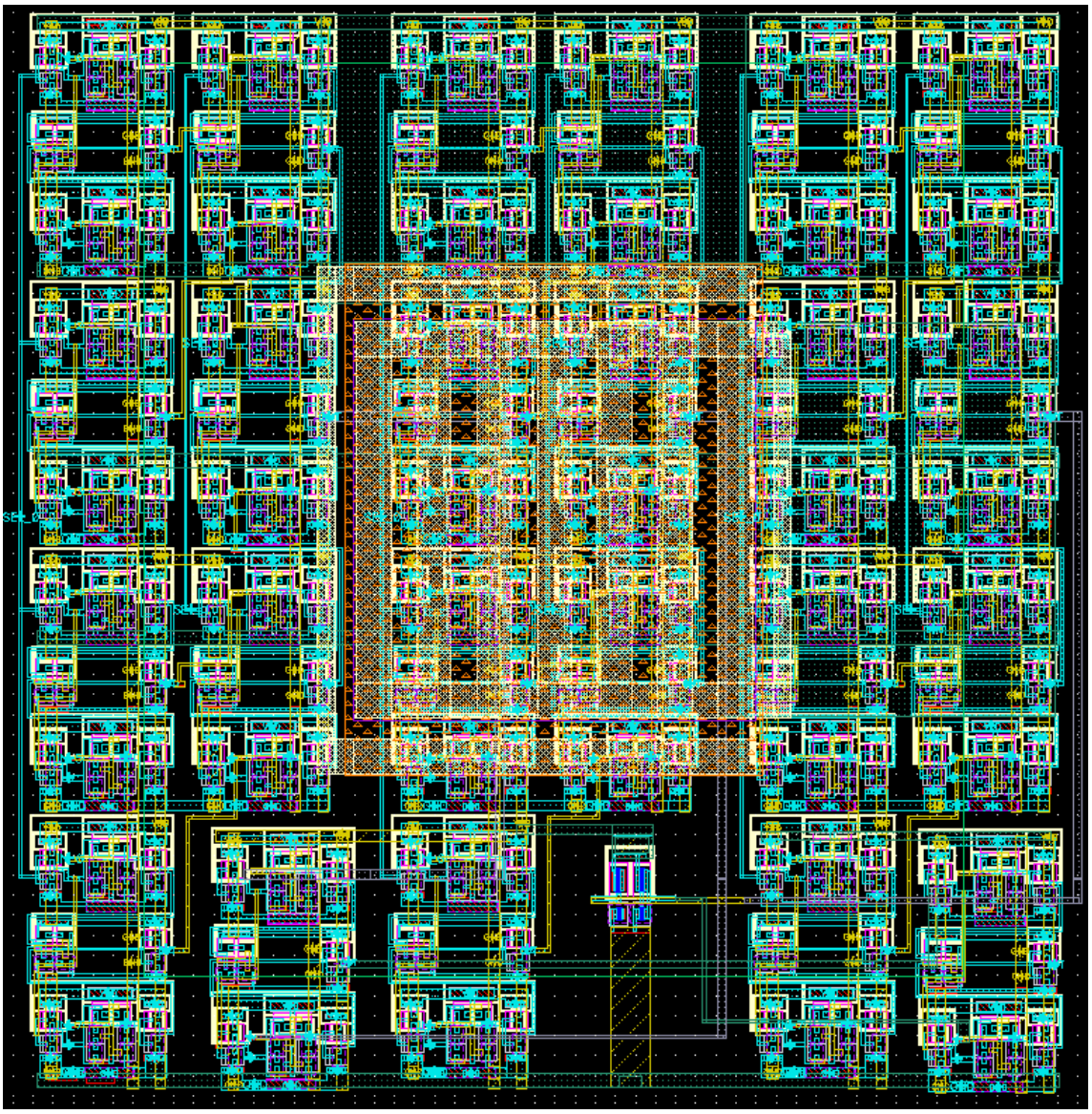

Figure 8.11. Layout of 24:1 MUX

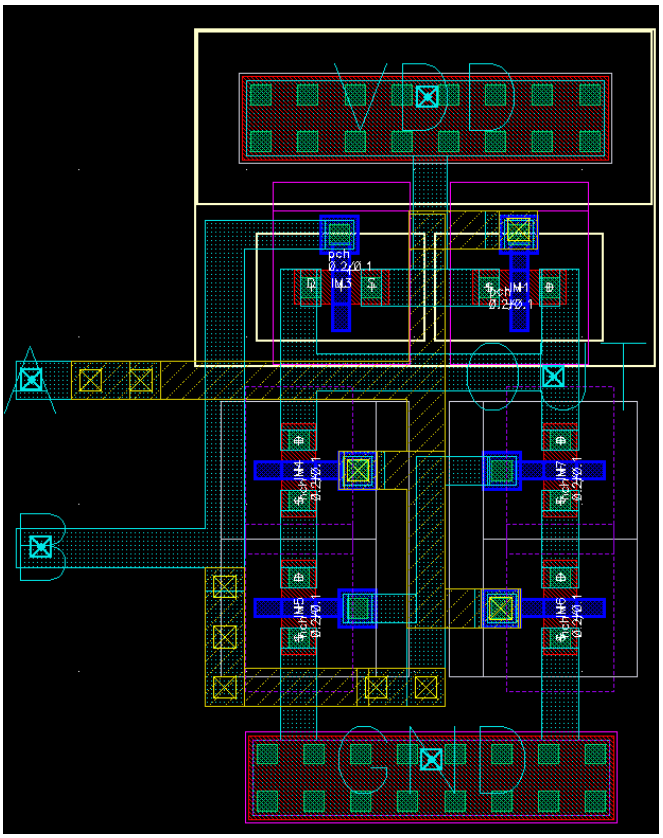

Figure 8.12. Layout of 2 input balanced NAND gate

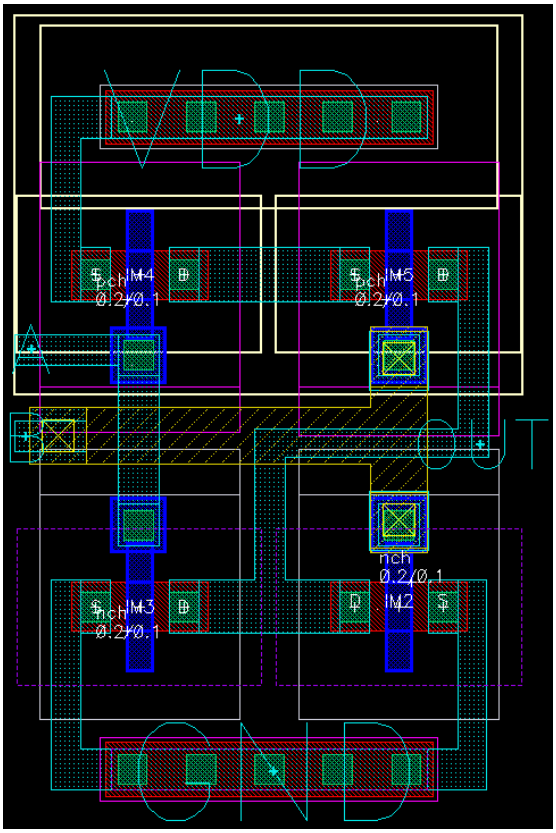

Figure 8.13. Layout of 2 input NOR gate

9 Appendix B: Schematics

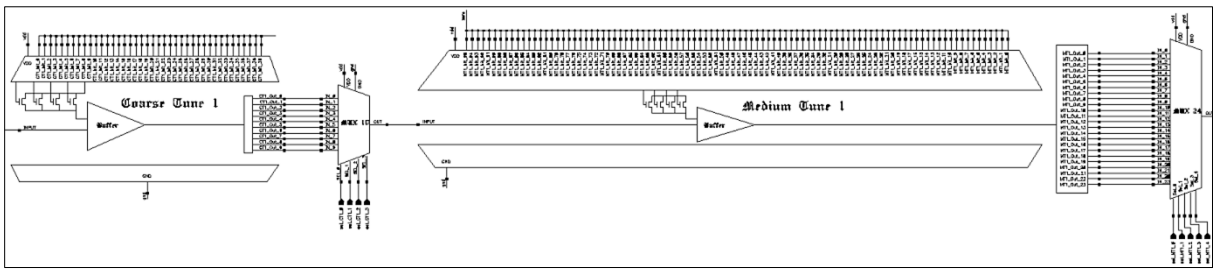

Figure 9.1. Schematics of Delay Line 1 (Current Starved)

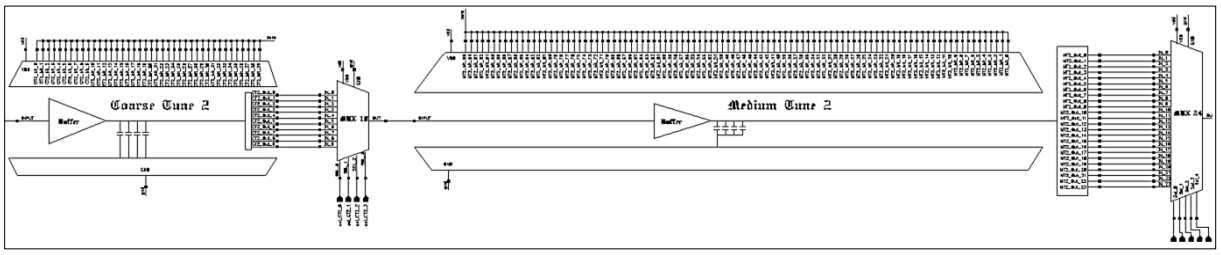

Figure 9.2. Schematics of Delay Line 2 (Shunt Capacitor)

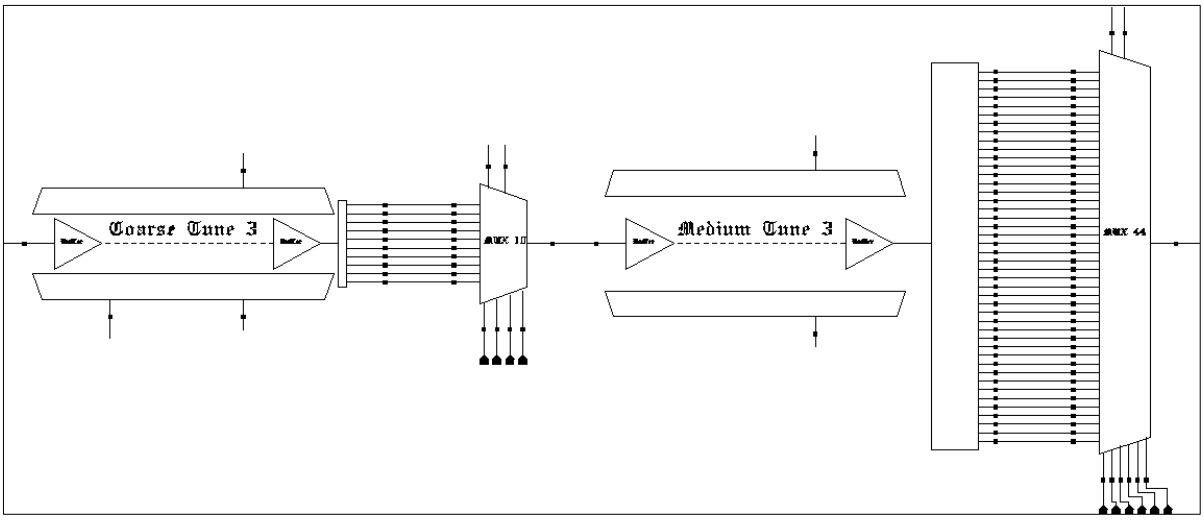

Figure 9.3. Schematics of Delay Line 3 (Non-Tunable buffers)

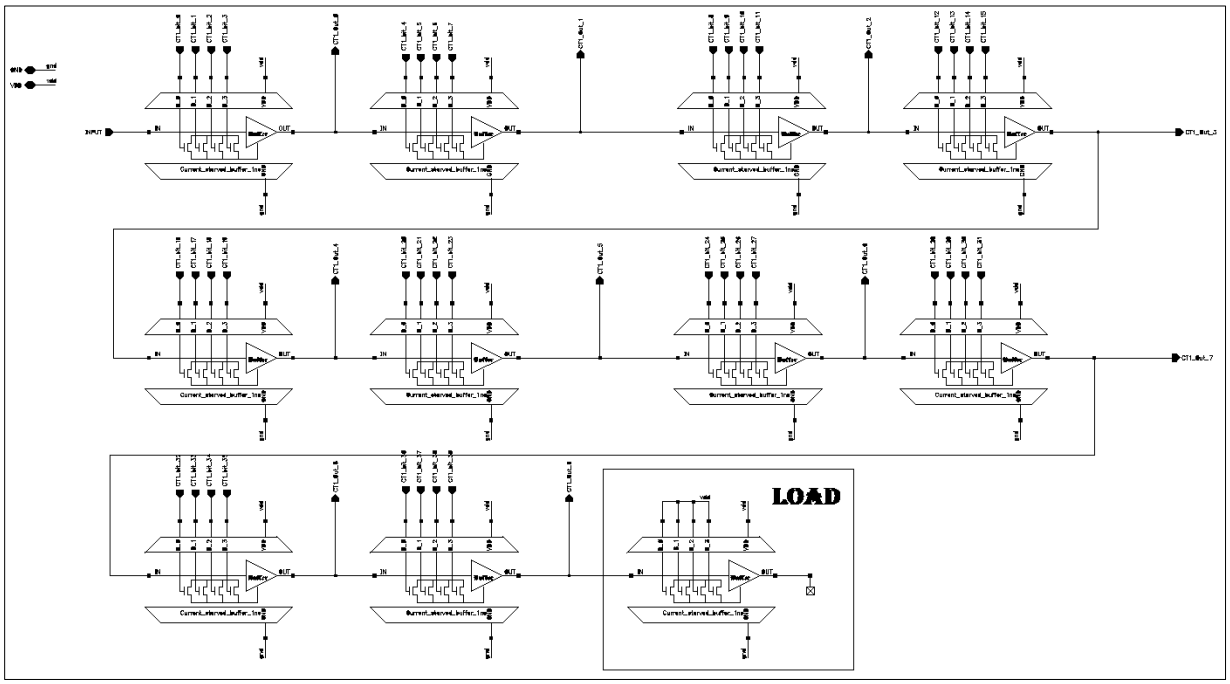

Figure 9.4. Schematics of Coarse tune 1 (Current Starved)

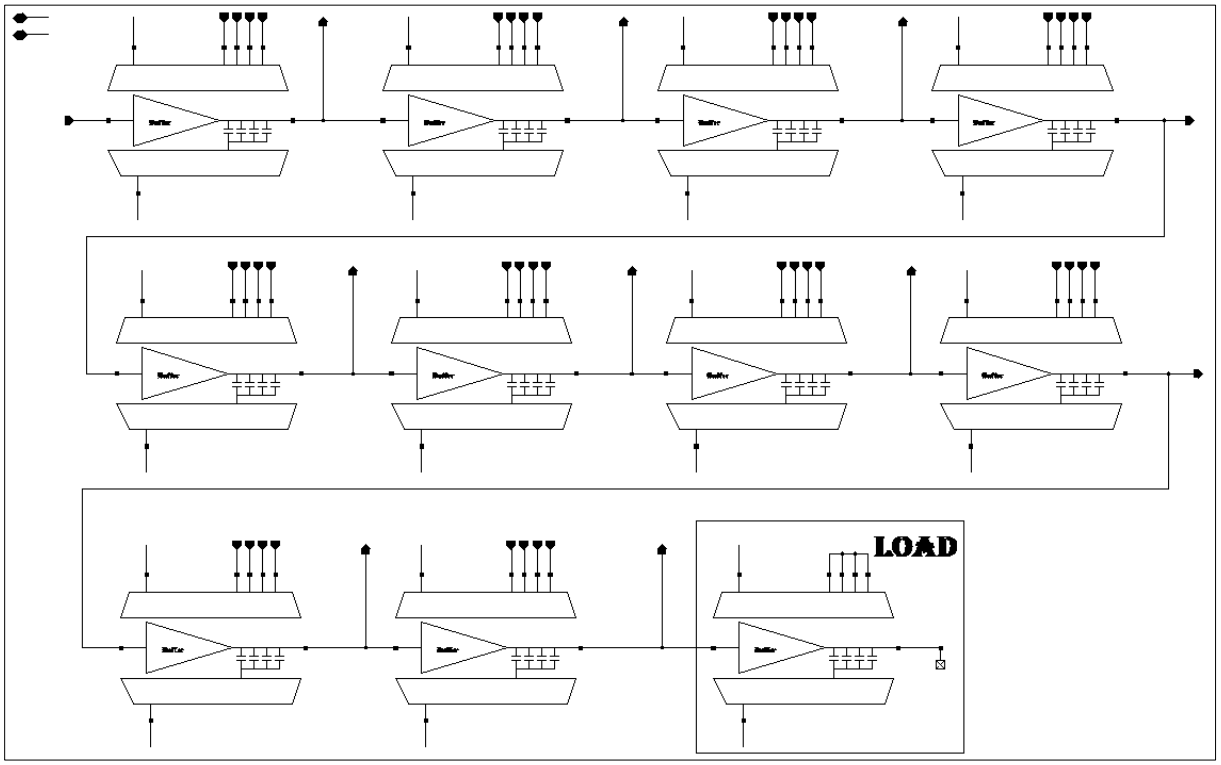

Figure 9.5. Schematics of Coarse Tune 2 (Shunt Capacitor)

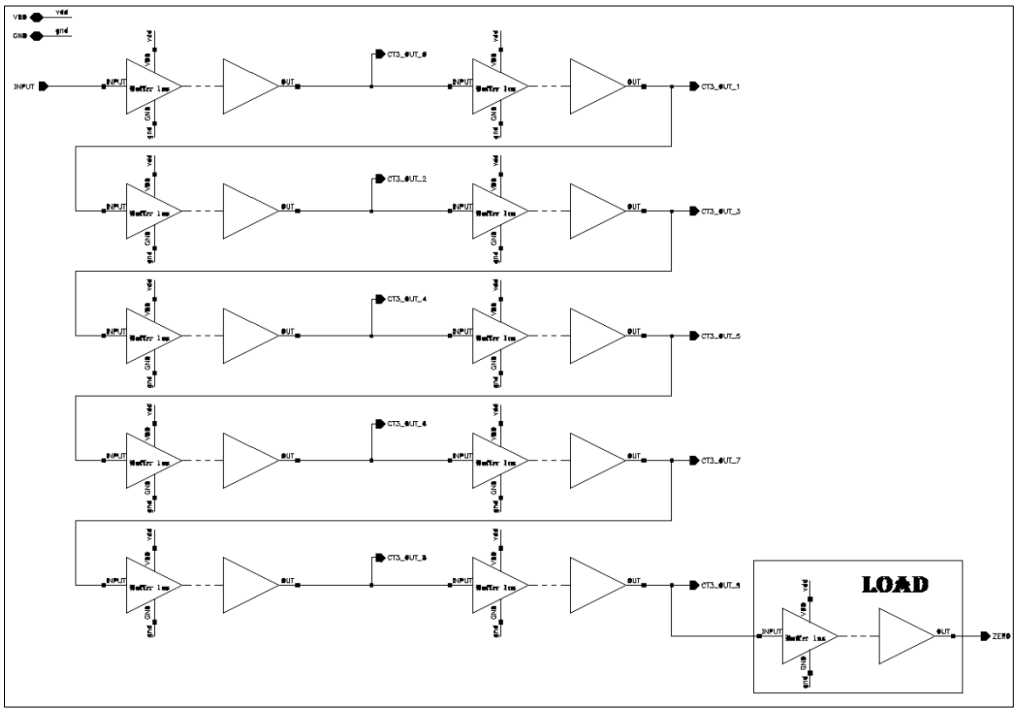

Figure 9.6. Schematics of Coarse Tune 3 (Non-Tunable Buffers)

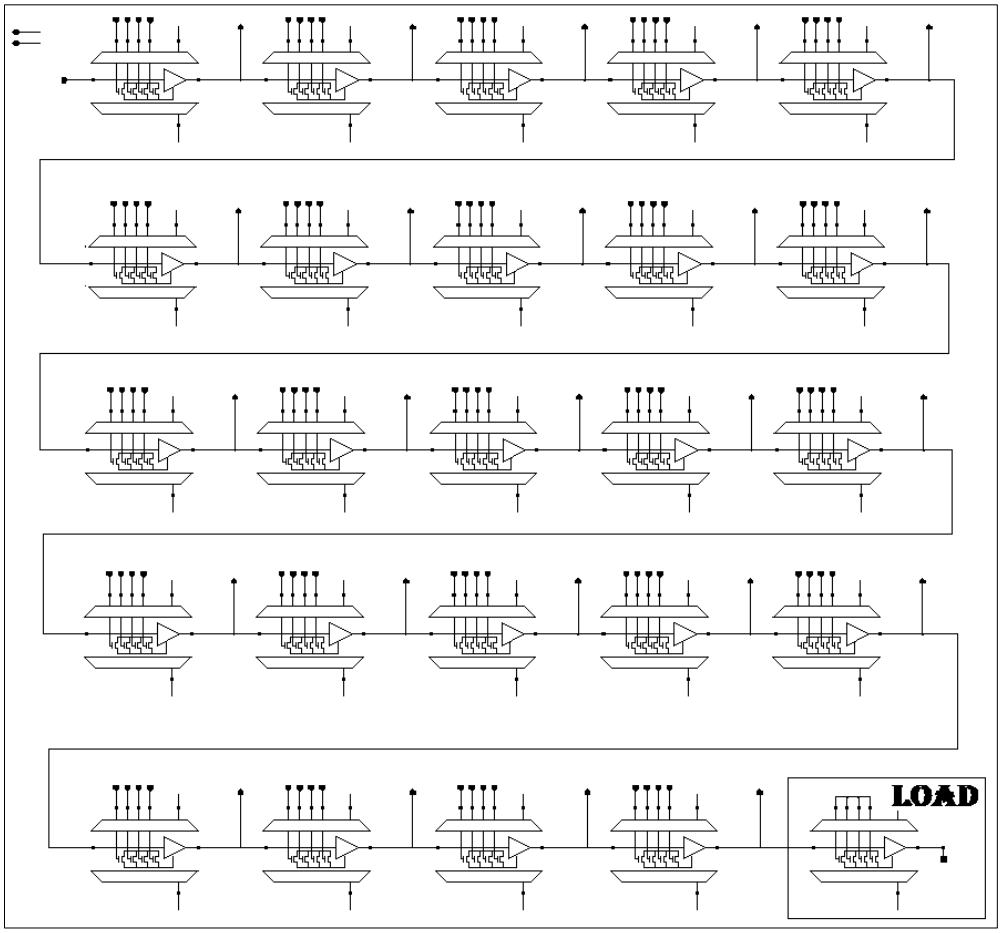

Figure 9.7. Schematics of Medium Tune 1 (Current Starved)

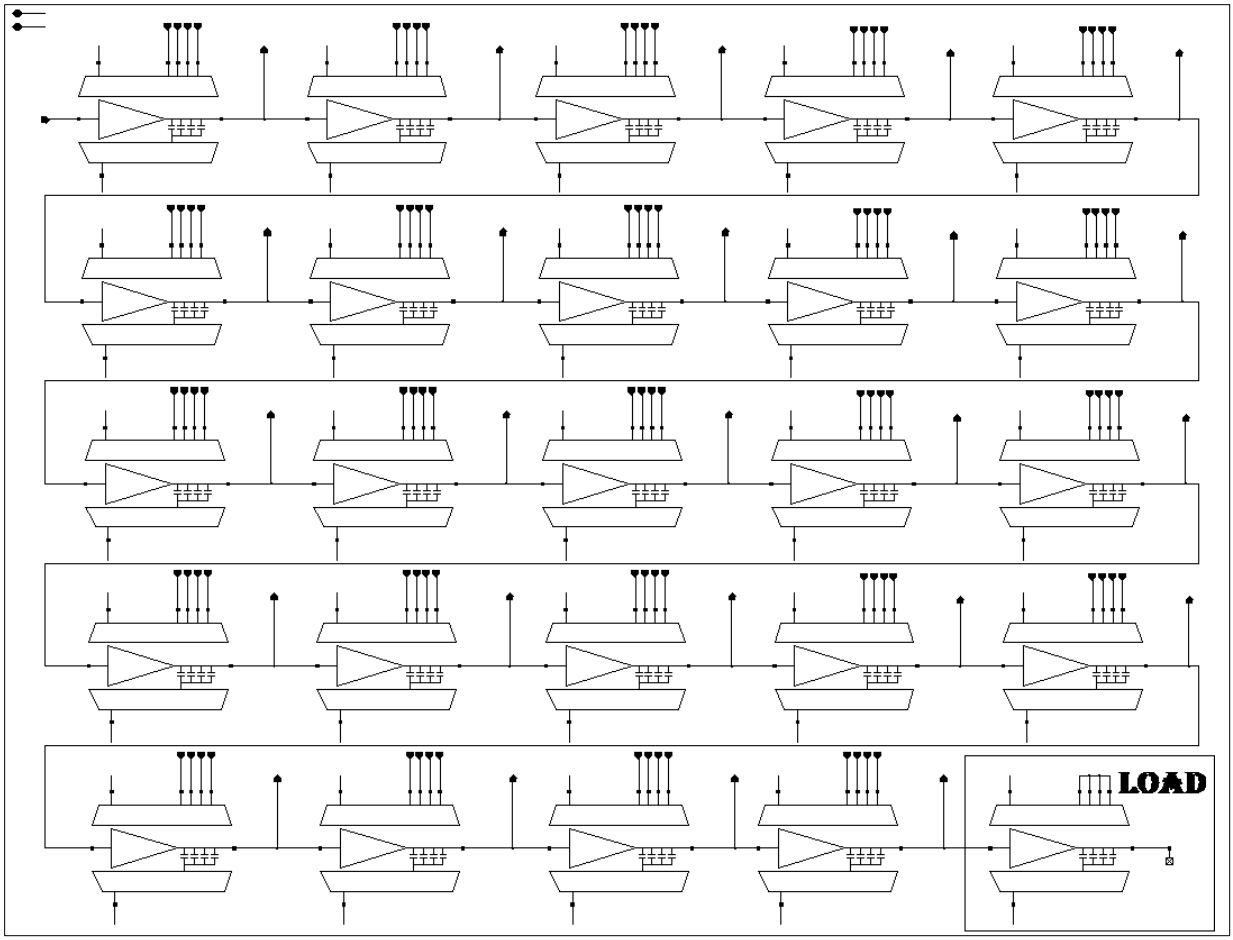

Figure 9.8. Schematics of Medium Tune 2 (Shunt Capacitor)

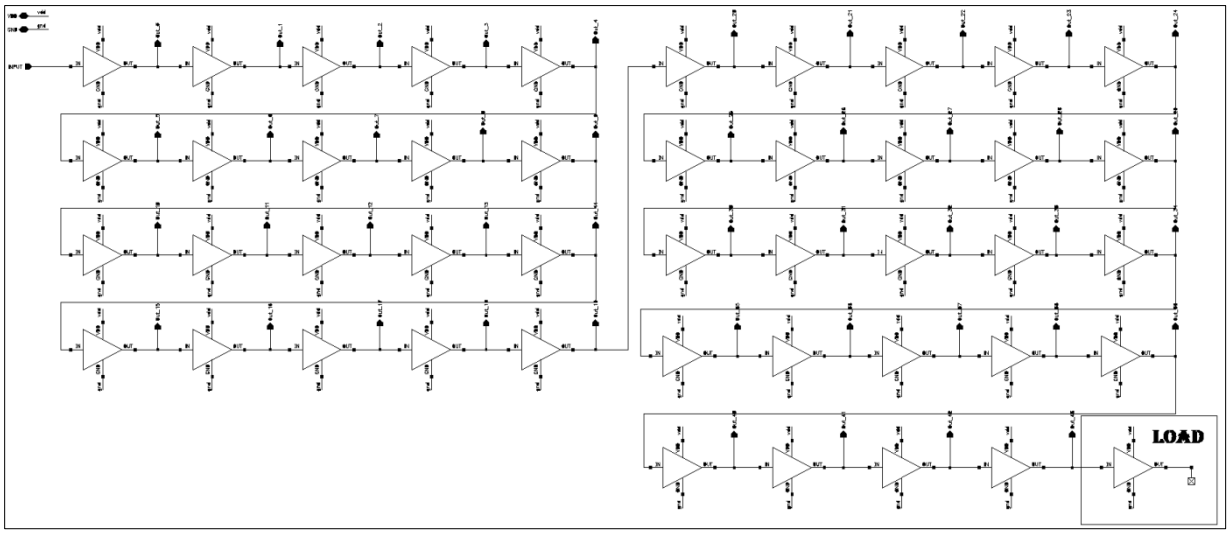

Figure 9.9. Schematics of Medium Tune 3 (Non-Tunable Buffers)

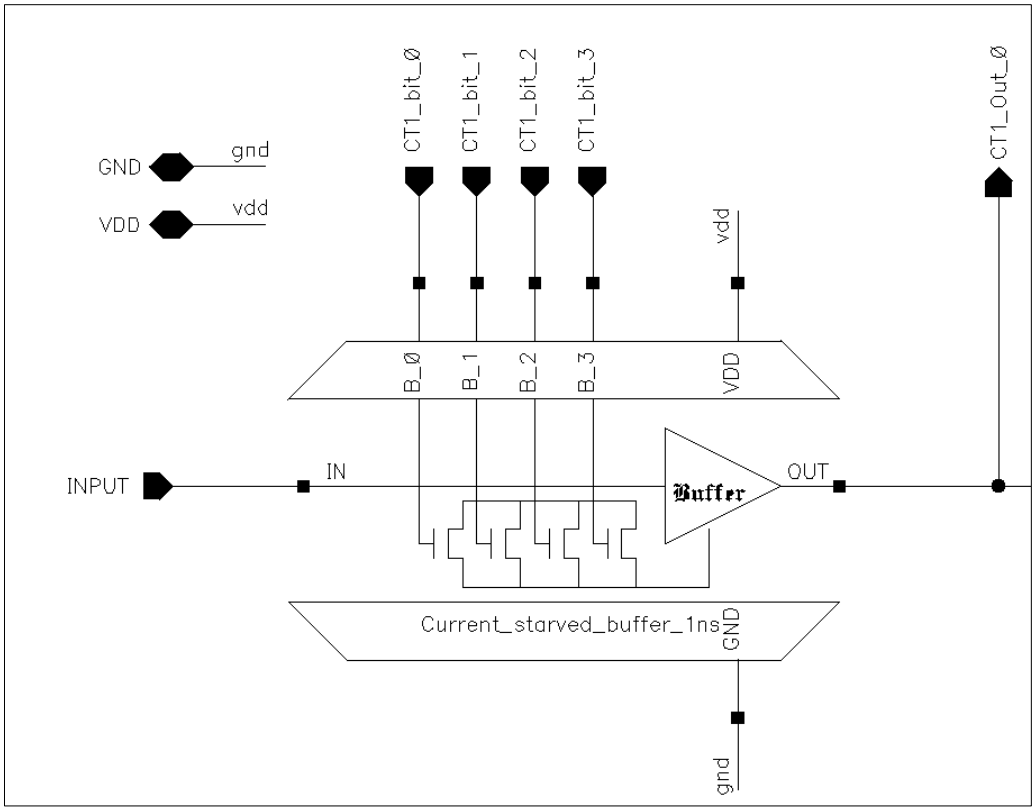

Figure 9.10. Schematics of Current Starved Coarse Tune Delay Element

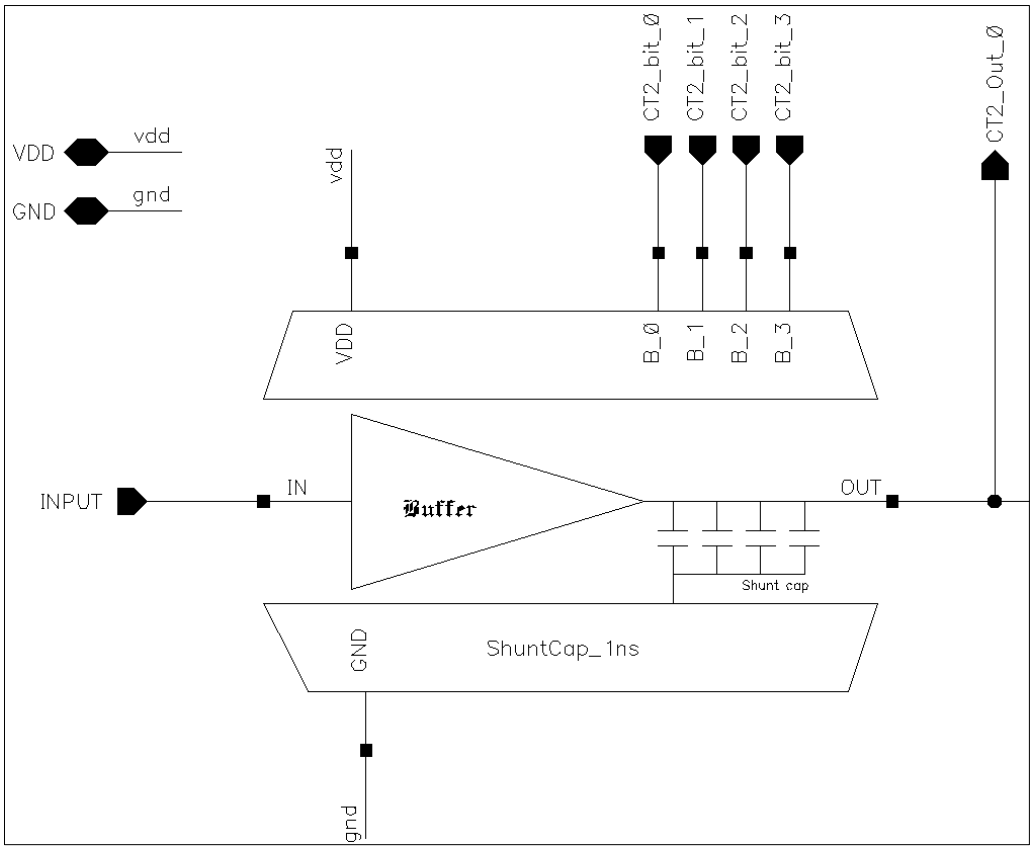

Figure 9.11. Schematics of Shunt Capacitor Coarse Tune Delay Element

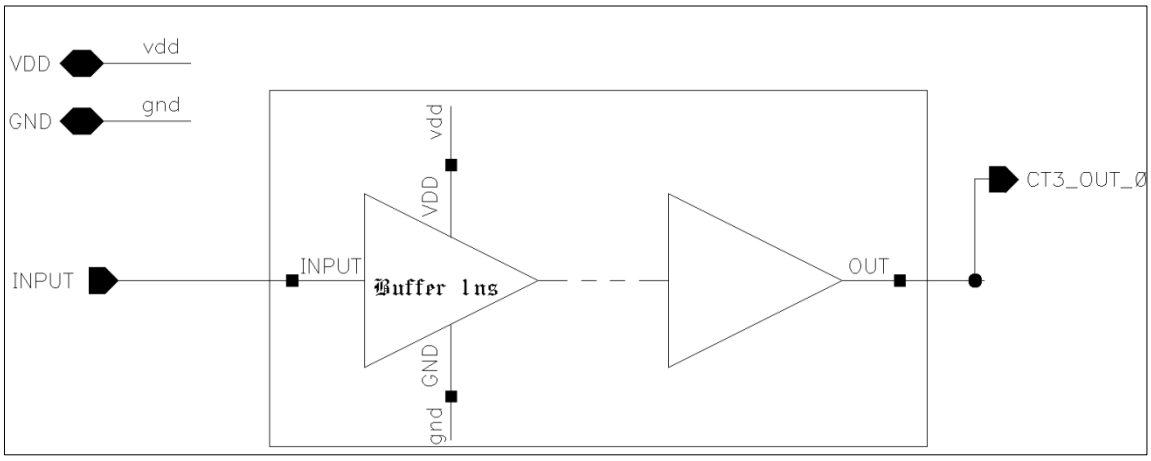

Figure 9.12. Schematics of Non-Tunable Buffer Coarse Tune Delay Element

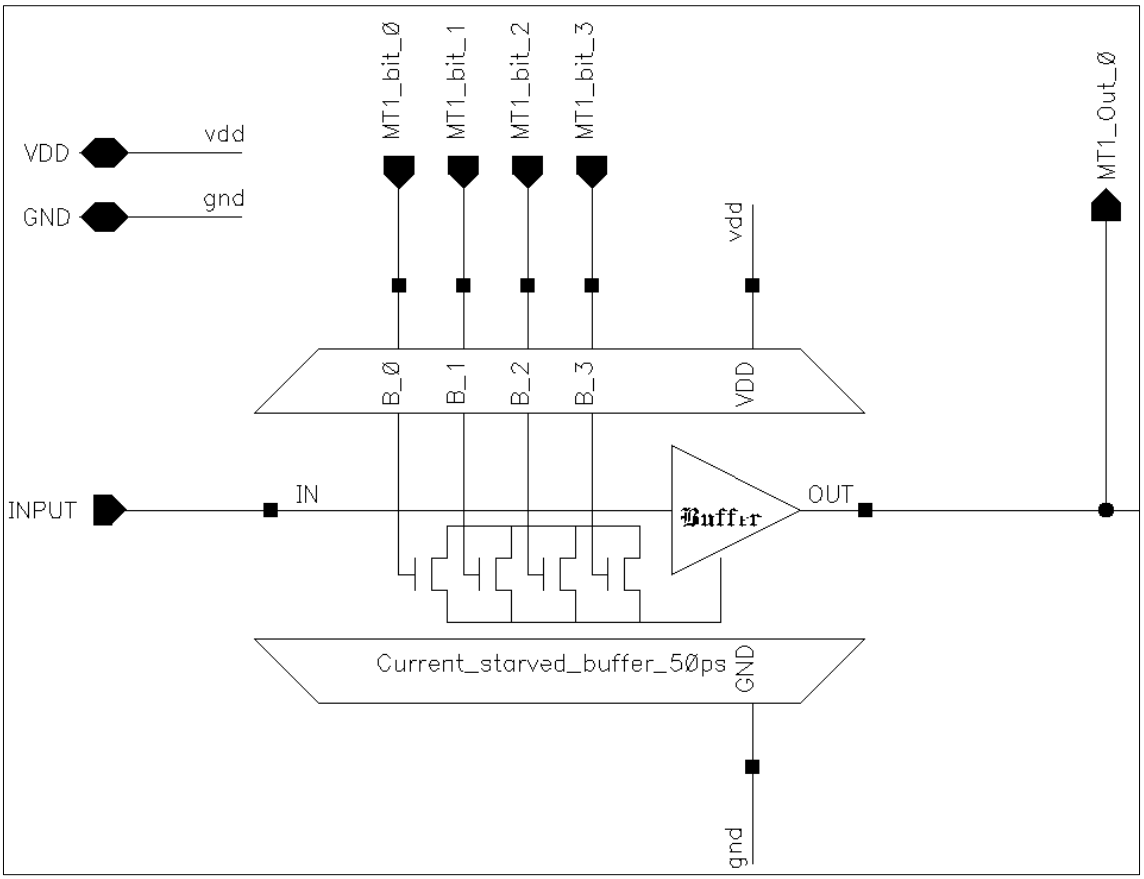

Figure 9.13. Schematics of Current Starved Medium Tune Delay Element

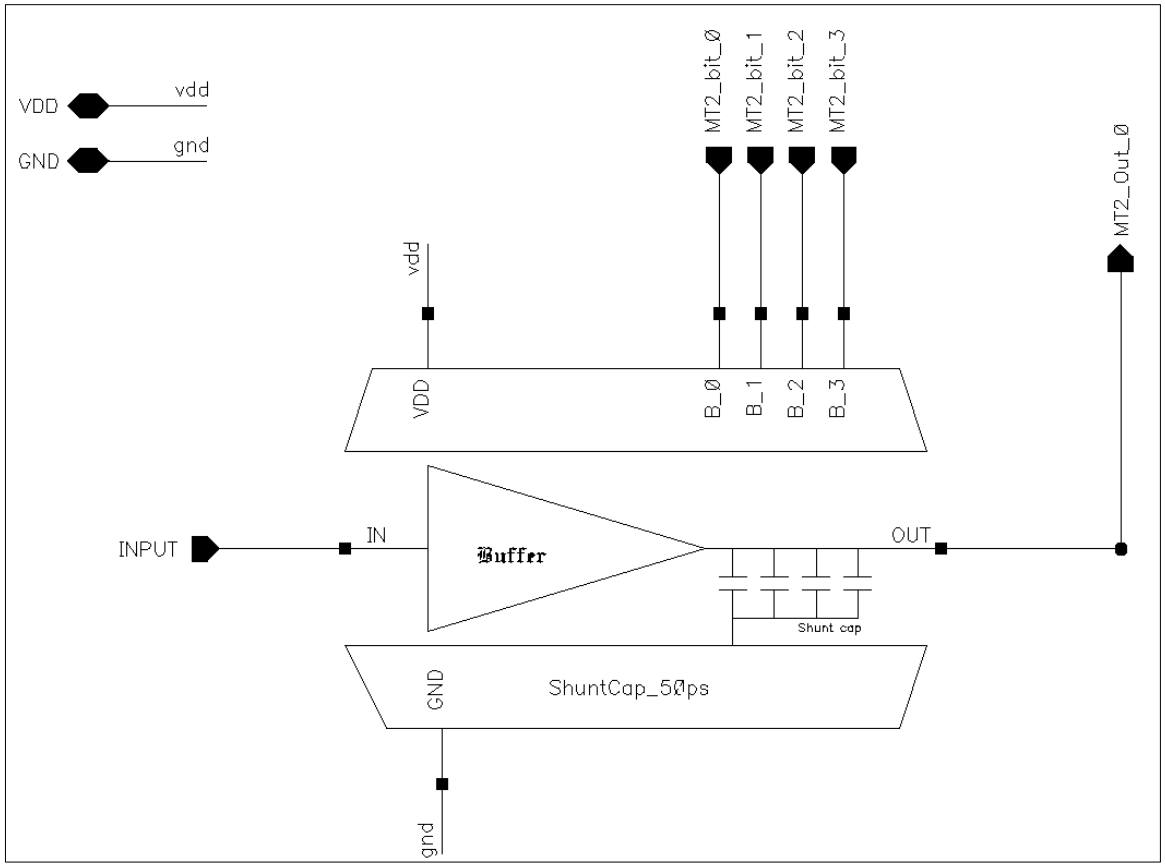

Figure 9.14. Schematics of Shunt Capacitor Medium Tune Delay Element

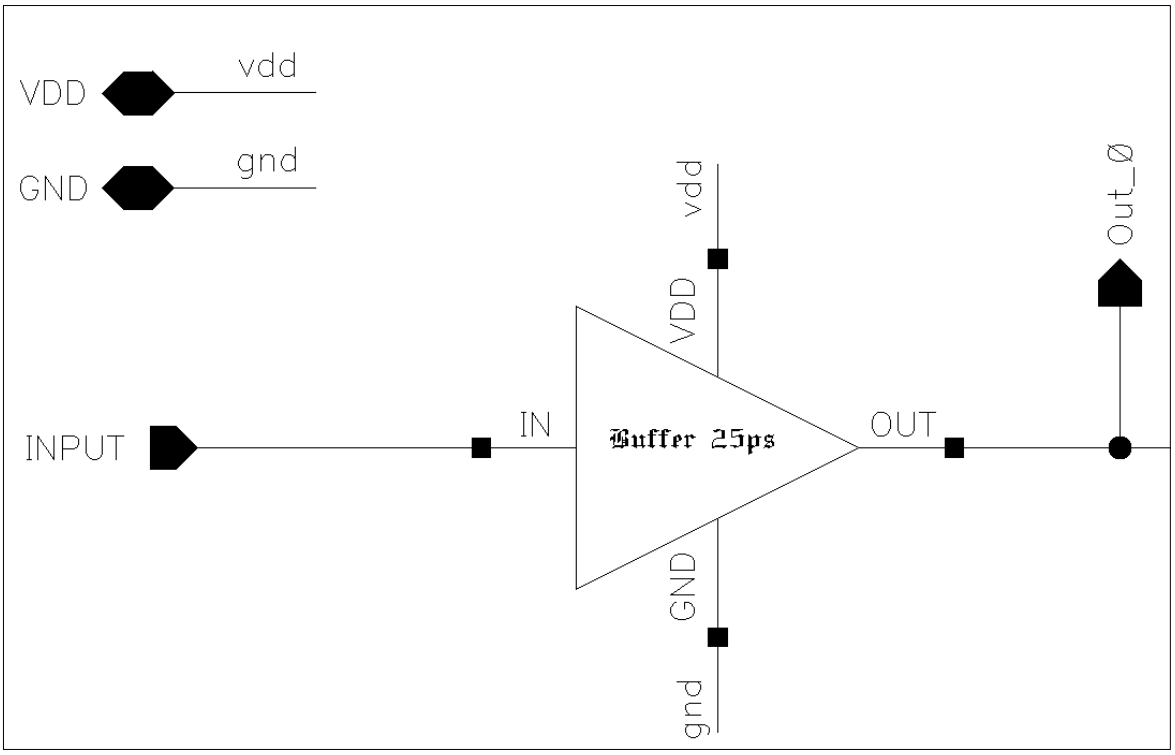

Figure 9.15. Schematics of Non-Tunable Buffer Medium Tune Delay Element

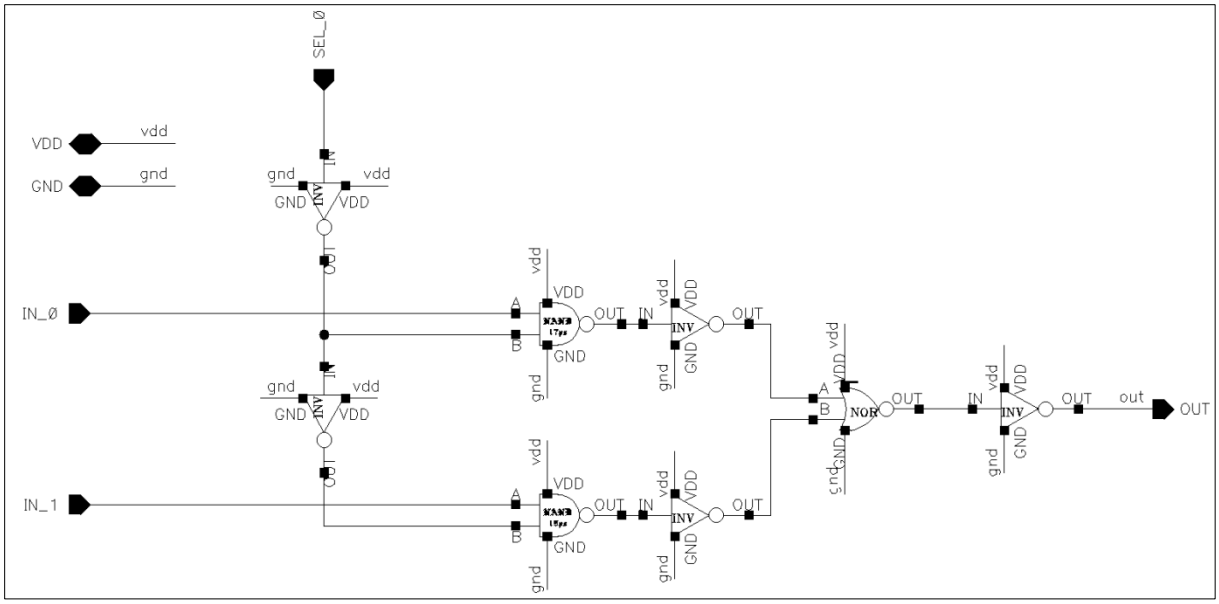

Figure 9.16. Schematics of 2:1 MUX

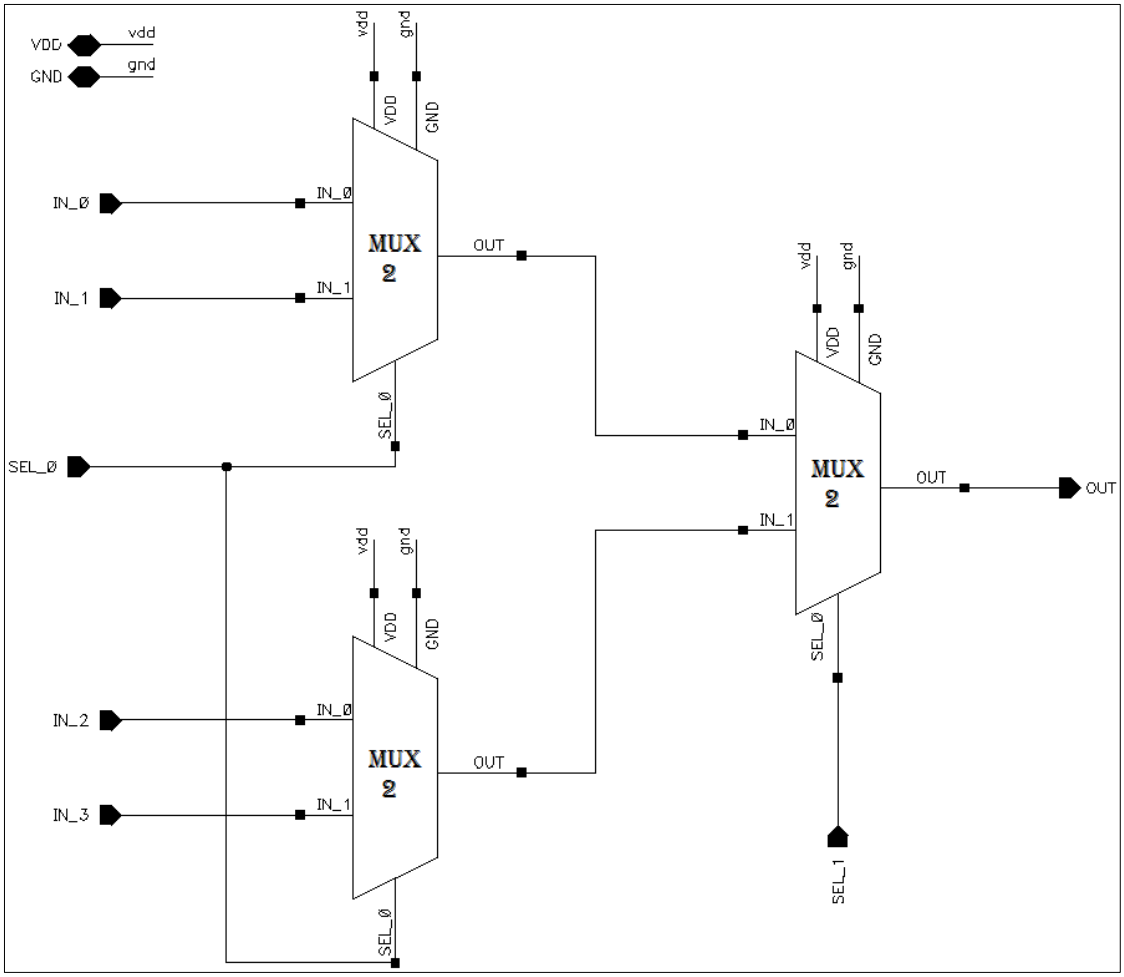

Figure 9.17. Schematics of 4:1 MUX

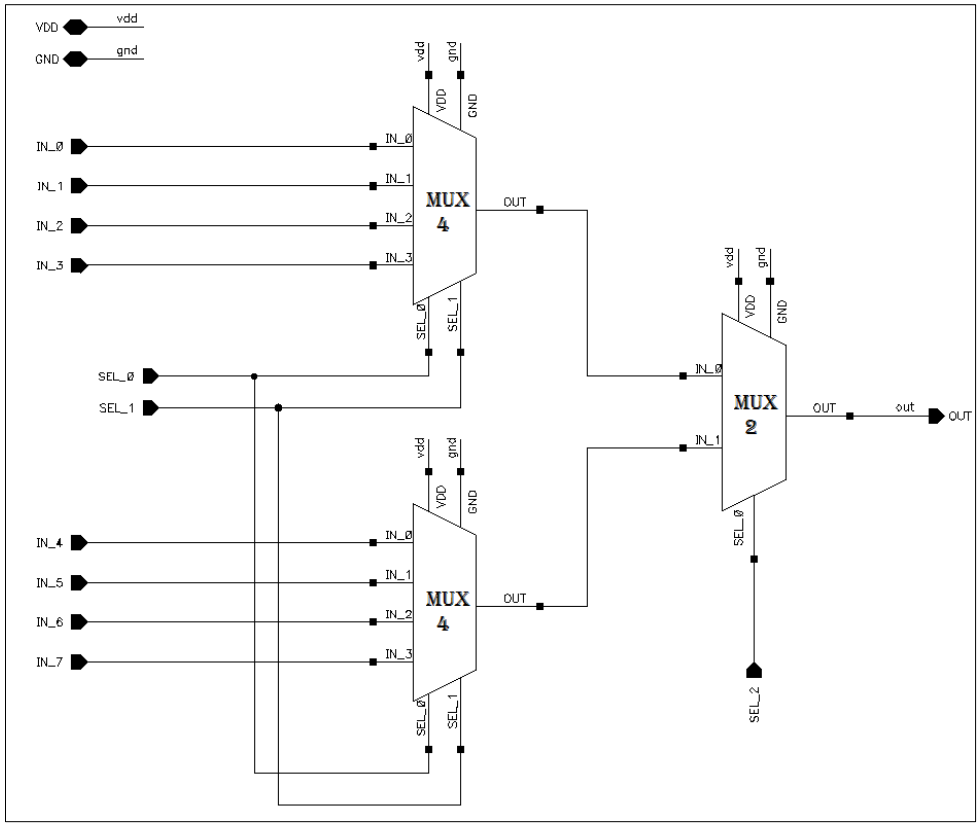

Figure 9.18. Schematics of 8:1 MUX

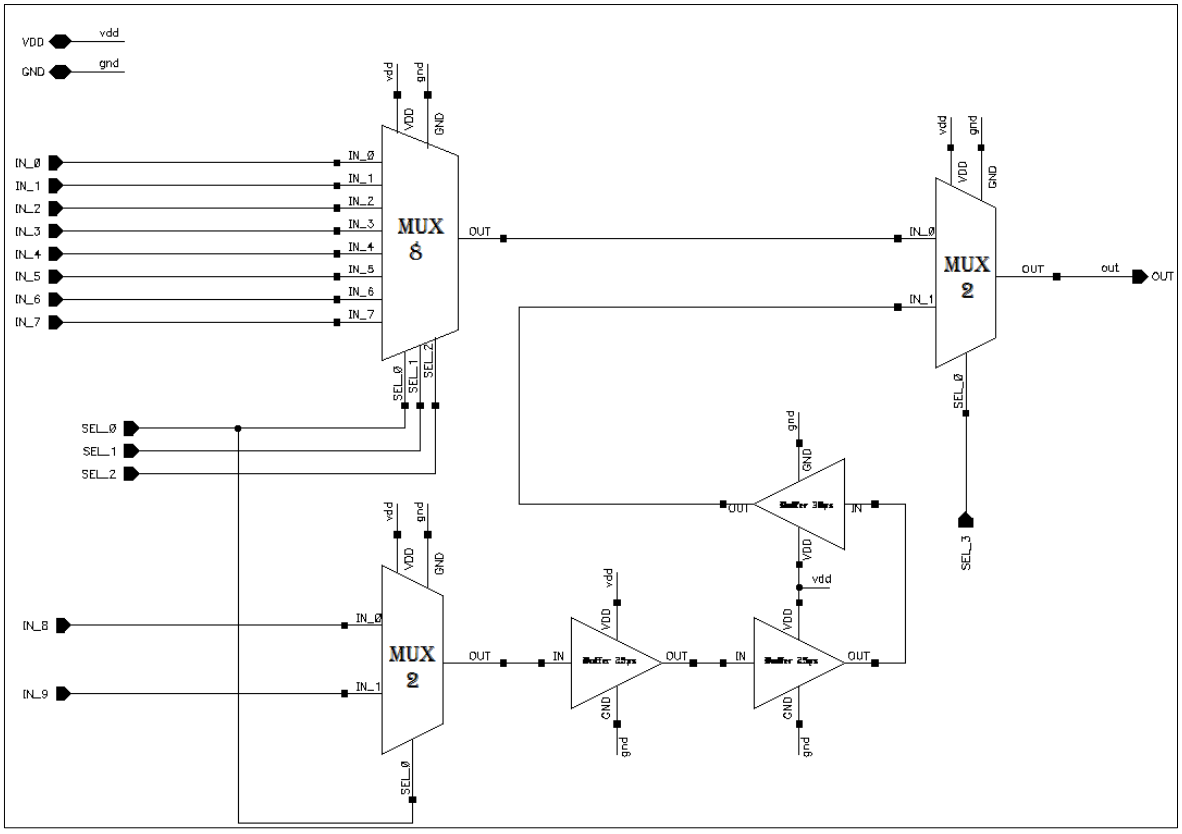

Figure 9.19. Schematics of 10:1 MUX

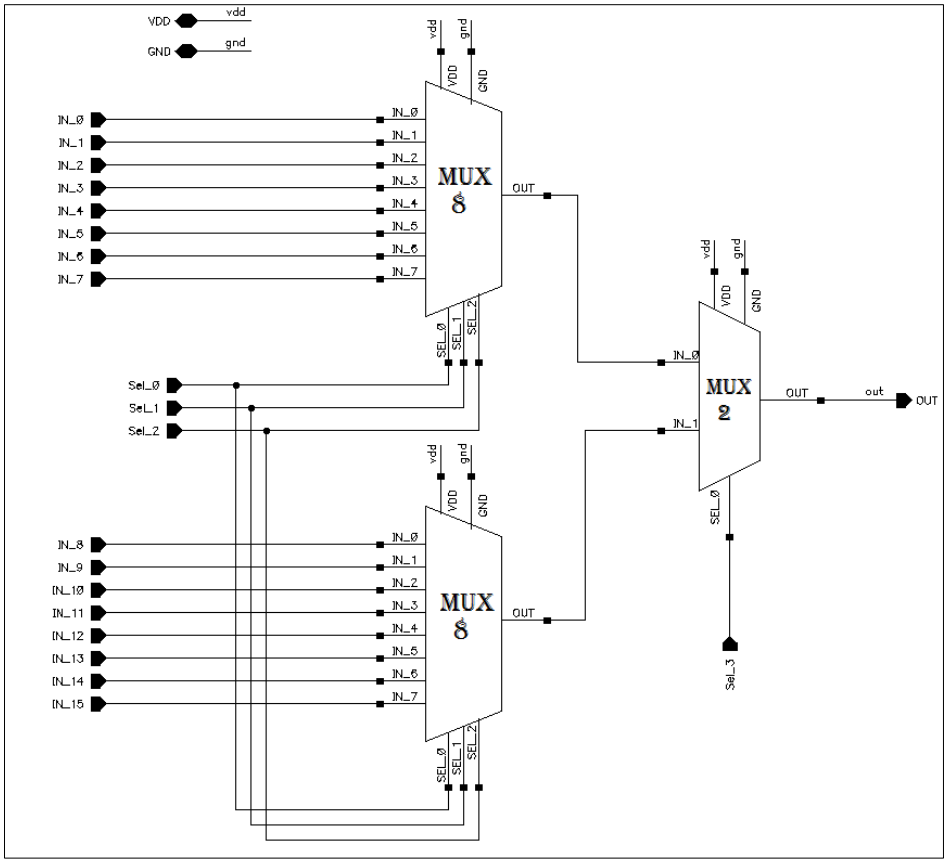

Figure 9.20 Schematics of 16:1 MUX

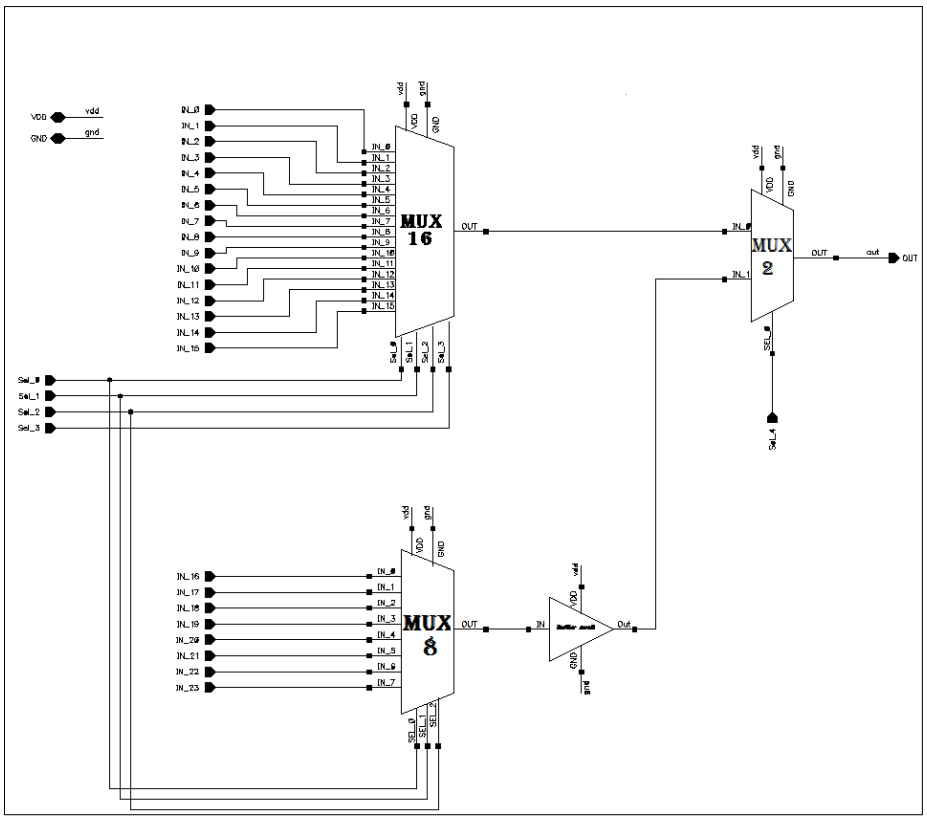

Figure 9.21 Schematics of 24:1 MUX

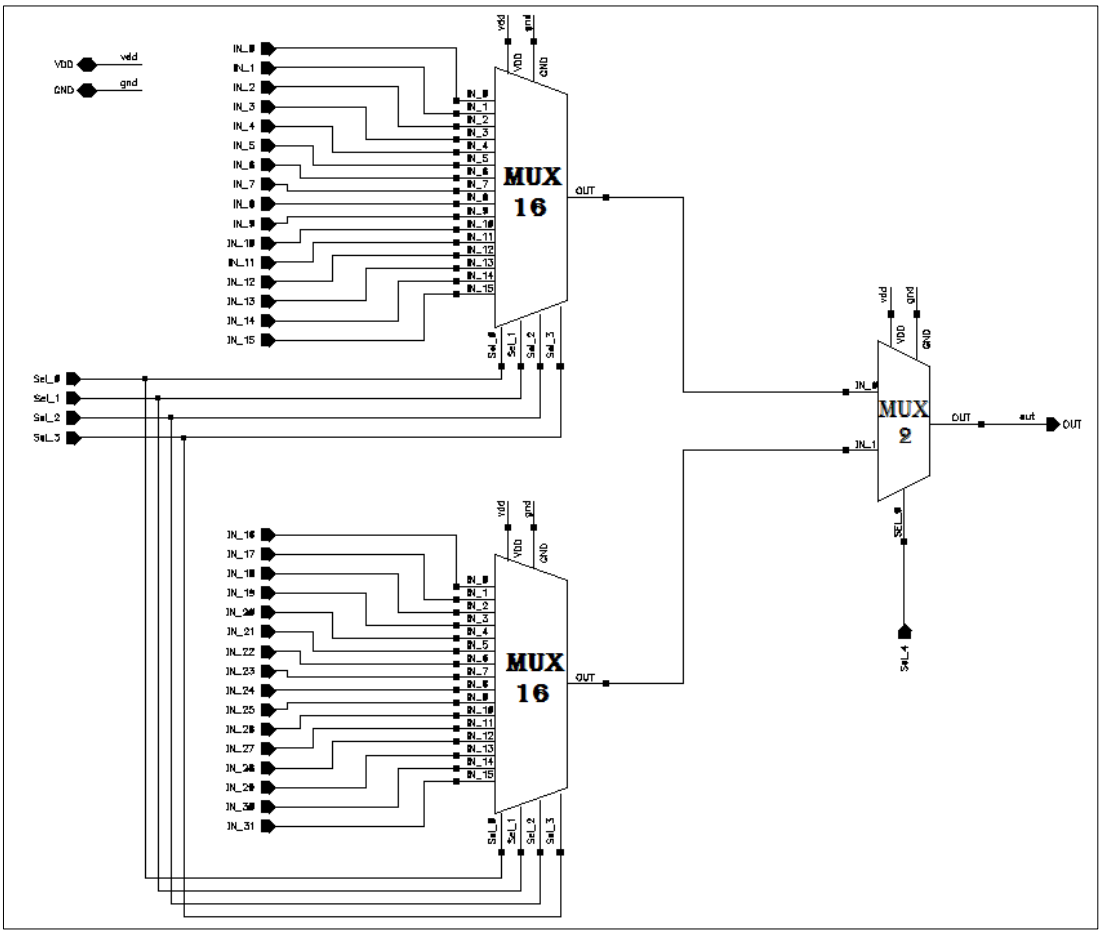

Figure 9.22 Schematics of 32:1 MUX

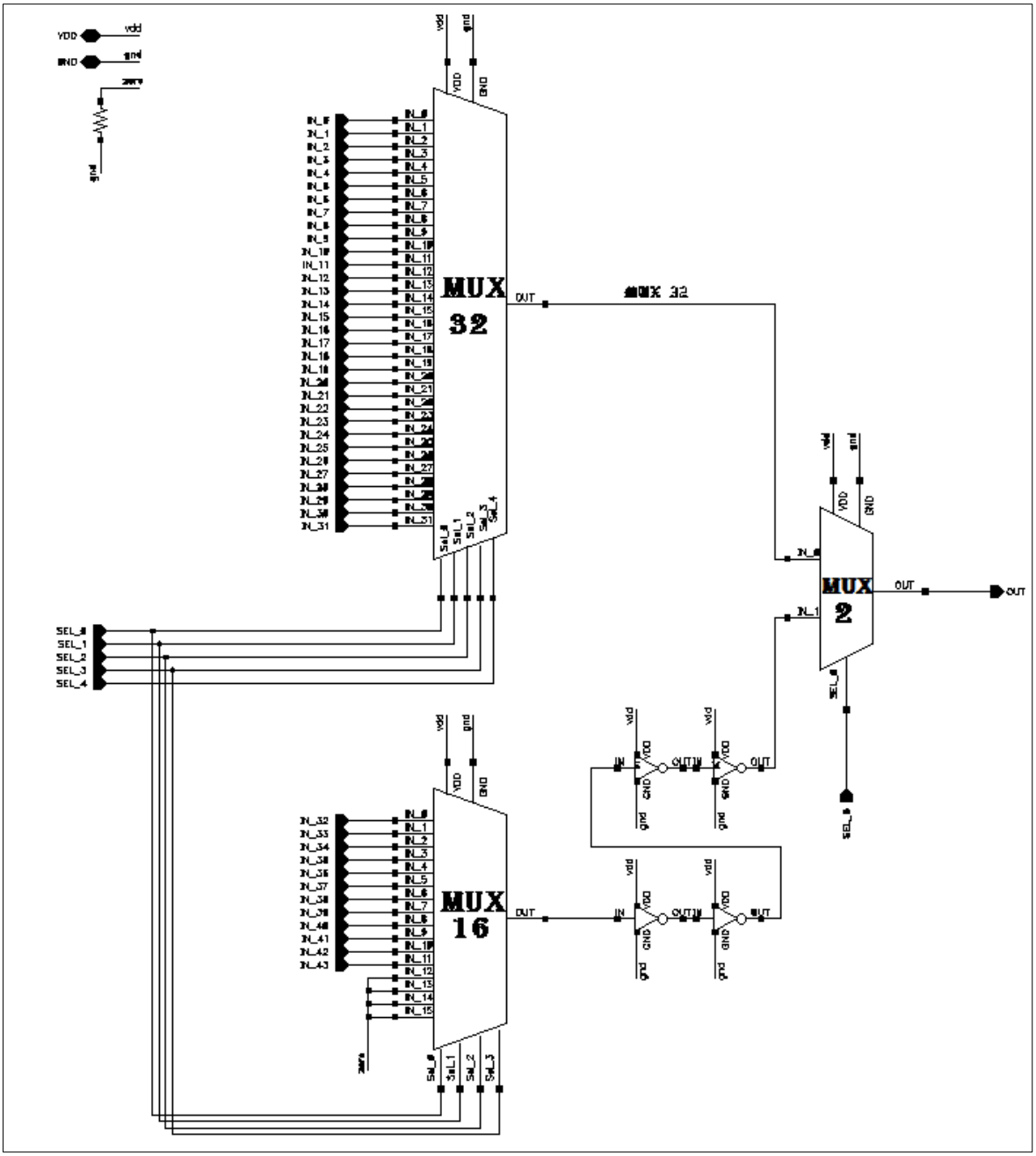

Figure 9.23 Schematics of 44:1 MUX

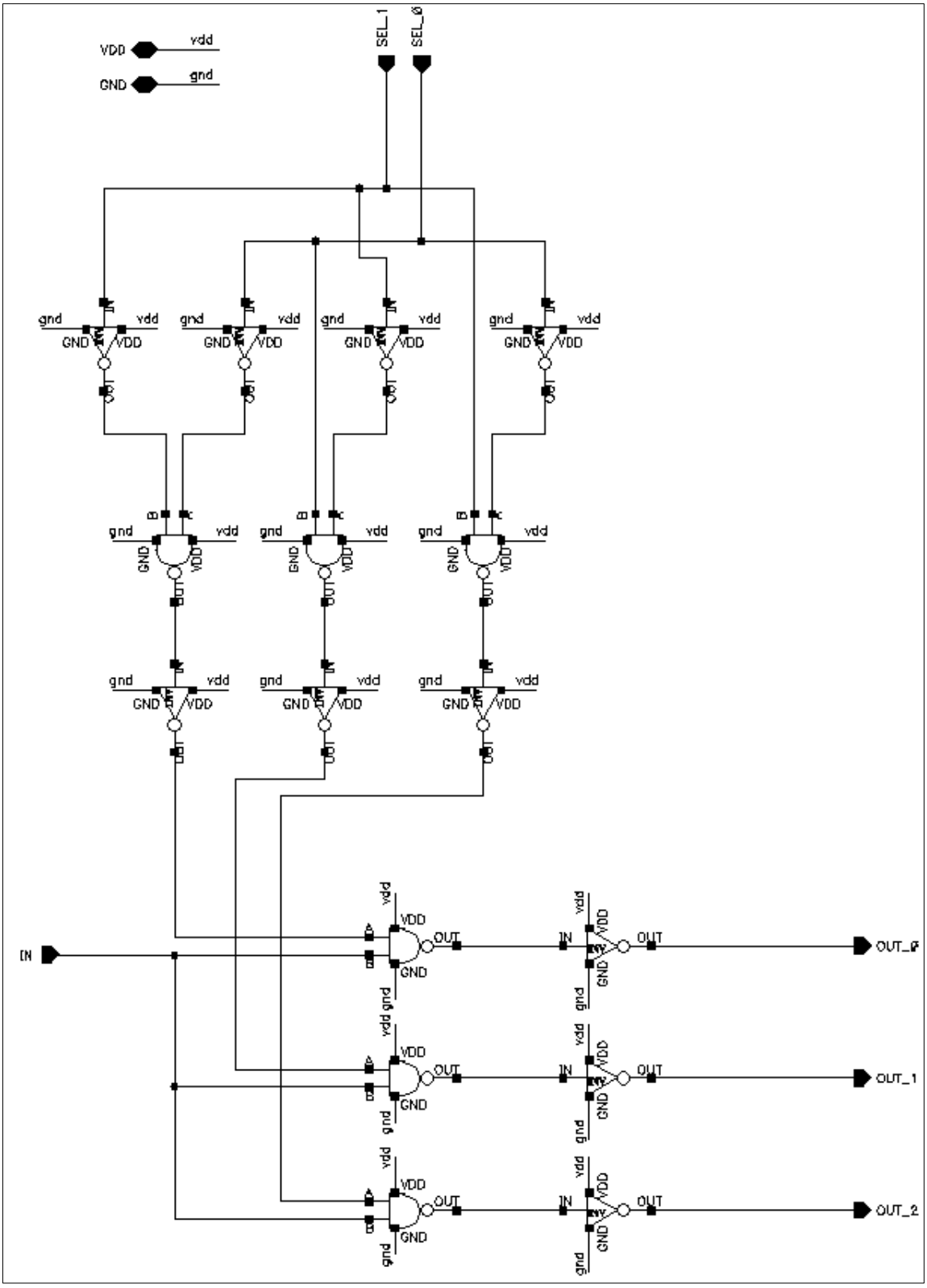

Figure 9.24 Schematics of 1:3 DMUX

10 List of Acronyms

ADE	Analog Design Environment
ASIC	Application Specific Integrated Circuit
ATD	Average Temperature Dependency
AVG	Average
BCT	Buffer Coarse Tune
BER	Bit Error Rate
BMT	Buffer Medium Tune
CC	Cycle-to-Cycle
CMOS	Complementary Metal-Oxide Semiconductor
CS	Current Starved
CSCT	Current Starved Coarse Tune
CSMT	Current Starved Medium Tune
CT	Coarse Tune
DC	Direct Current
DCDE	Digitally Controllable Delay Elements
DCO	Digitally Controlled Oscillator
DLL	Delay-Locked Loop
DNL	Differential Non-Linearity
DUT	Device Under Test
EMI	Electro Magnetic Interference
FTP	Foiled Twisted Pair
JFET	Junction gate Field Effect Transistor
LSB	Least Significant Bit
MC	Monte Carlo
MOSFET	Metal-Oxide Semiconductor Field Effect Transistor
MT	Medium Tune
nMOS	n-type MOS Transistor
NTB	Non-Tunable Buffer
OPAMP	Operational Amplifier
PEX	Parasitic Extraction and Simulation
PLL	Phase Locked Loop
pMOS	p-type MOS Transistor
Pnoise	Periodic Noise Analysis
PRF	Pulse Repetition Frequency
PSS	Periodic Steady State Analysis
PVT	Process Voltage and Temperature

PWR	Power
RAD-Hard	Radiation Hardening
RAM	Random Access Memory
RMS	Root Mean Square
RX	Receiver
SC	Shunt Capacitor
SCCT	Shunt Capacitor Coarse Tune
SCDM	Serial Communications Device Manufacturers
SCMT	Shunt Capacitor Medium Tune
SPICE	Simulation Program with Circuit Emphasis
STP	Shielded Twisted Pair
TP	Twisted Pair
TX	Transmitter
VTCMOS	Variable Threshold Complementary Metal-Oxide Semiconductor

11 Bibliography

[1] Maymandi-Nejad, M.; Sachdev, M.; , *"A digitally programmable delay element: design and analysis",* Very Large Scale Integration (VLSI) Systems, IEEE Transactions on , vol.11, no.5, pp.871-878, Oct. 2003

[2] Masaharu Goto, James O. Barnes and Ronnie E. Owens; "*CMOS Programmable Delay Vernier",* Hewlett-Packard Journal, pp. 51-58, October 1994.

[3] David A. Johns; Ken Martin; "*Analog Integrated Circuit Design"*, p.106, John Wiley & Sons, Inc., 1997.

[4] Art Zirger; *"Random Offset in CMOS IC Design"*, ECEN4827/5827 Analog IC Design, p.9, National Semiconductor, October 19, 2007.

[5] Balasubramanian Narasimhan; "*The Normal Distribution"*, Department of Statistics, Stanford, Jul. 22, 1996.

[6] Bruce W. Smith; Hoyoung Kang; Anatoly Bourov; Frank Cropanese; Yongfa Fan; *"Water Immersion Optical Lithography for the 45nm Node"*, Rochester Institute of Technology, Microelectronic Engineering Department , July 2004.

[7] Art Zirger; *"Random Offset in CMOS IC Design"*, ECEN4827/5827 Analog IC Design, p.10, National Semiconductor, October 19, 2007.

[8] David Harris; *"Introduction to CMOS VLSI Design; Lecture 15: Non-ideal Transistors"*, Harvey Mudd College, p.24. Spring 2004.

[9] N. Weste; D.Harris; "*MOS Transistor Theory"*, CMOS VLSI Design, Addison-Wesley, 3/e, Eastern Washington University, p.36, 2004.

[10] Masaharu Goto; James O. Barnes; Ronnie E. Owens; "*CMOS Programmable Delay Vernier",* Hewlett-Packard Journal, p. 54, October 1994.

[11] David A. Johns; Ken Martin; "*Analog Integrated Circuit Design"*, p.199, John Wiley & Sons, Inc., 1997.

[12] Figueiredo, M.J.; Aguiar, R.L.; , "Noise and Jitter in CMOS Digitally Controlled Delay Lines," Electronics, Circuits and Systems, 2006. ICECS '06. 13th IEEE International Conference, vol., no., pp.1356-1359, 10-13 Dec. 2006

[13] Agilent Technologies; *"Measuring Jitter in Digital Systems"*, Application Note 1448-1, p.4, 2008.

[14] Jie Sun; Li, M.; Wilstrup, J.; , *"A demonstration of deterministic jitter (DJ) deconvolution,"* Instrumentation and Measurement Technology Conference, 2002. IMTC/2002. Proceedings of the 19th IEEE , vol.1, no., pp. 293- 298 vol.1, 2002

[15] Johnnie Hancock; "*Finding Sources of Jitter with Real-Time Jitter Analysis"*, Agilent Technologies, Application Note 1448-2, p.1, March 25. 2008.

[16] Dr. Howard Johnson; *"Random and Deterministic Jitter"*, First publ. in EDN Magazine, p.1, June 27, 2002.

[17] C.D. Motchenbacher; J. A. Connelly; "*Low-Noise Electronic System Design"*, John Wiley & Sons, INC, pp. 18-21, 1993.

[18] Silicon Laboratories; "*Calculating Total Output Jitter for PLL's"*, Technical Documents, AN56, pp.1-6, 2012.

[19] Figueiredo, M.J.; Aguiar, R.L.; , "Noise and Jitter in CMOS Digitally Controlled Delay Lines," Electronics, Circuits and Systems, 2006. ICECS '06. 13th IEEE International Conference, vol., no., pp.1358, 10-13 Dec. 2006

[20] Craig Emmerich; "*Introduction to Jitter"*, Wavecrest, RAMBUS Developer Forum, p.4, October 23. 2001.

[21] Craig Emmerich; "*Introduction to Jitter"*, Wavecrest, RAMBUS Developer Forum, p.19, October 23. 2001.

[22] Johnnie Hancock; "*Jitter – Understanding it, Measuring it, Eliminating it Part 1: Jitter Fundamentals"*, High Frequency Electronics, Summit Technical Media, LLC, p.45, April 2004.

[23] Agilent Technologies; "*Analyzing Data Using Eye Diagrams",* Agilent Network Analyzer Service & Support Home Page, Oct. 21. 2011.

[24] C.D. Motchenbacher; J. A. Connelly; "*Low-Noise Electronic System Design"*, John Wiley & Sons, INC, pp. 8-11, 1993.

[25] Johnnie Hancock; "*Finding Sources of Jitter with Real-Time Jitter Analysis"*, Agilent Technologies, Application Note 1448-2, p.4, March 25. 2008.

[26] C.D. Motchenbacher;J. A. Connelly; "*Low-Noise Electronic System Design"*, John Wiley & Sons, INC, p. 10, 1993.

[27] C.D. Motchenbacher; J. A. Connelly; "*Low-Noise Electronic System Design"*, John Wiley & Sons, INC, pp. 30-31, 1993.

[28] Figueiredo, M.J.; Aguiar, R.L.; , "Noise and Jitter in CMOS Digitally Controlled Delay Lines," Electronics, Circuits and Systems, 2006. ICECS '06. 13th IEEE International Conference, vol., no., pp.1359, 10-13 Dec. 2006

[29] C.D. Motchenbacher; J. A. Connelly; "*Low-Noise Electronic System Design"*, John Wiley & Sons, INC, pp. 25-27,1993.

[30] Jack Browne; *"Integrated System Tests Device Flicker Noise", MicroWaves&RF, Jul. 15, 2008.*

[31] C.D. Motchenbacher; J. A. Connelly; "*Low-Noise Electronic System Design"*, John Wiley & Sons, INC, p. 150, 1993.

[32] C.D. Motchenbacher; J. A. Connelly; "*Low-Noise Electronic System Design"*, John Wiley & Sons, INC, p. 28, 1993.

[33] Sudo, T.; Sasaki, H.; Masuda, N.; Drewniak, J.L.; *, "Electromagnetic interference (EMI) of system-on-package (SOP),"* Advanced Packaging, IEEE Transactions on , vol.27, no.2, pp. 304- 314, May 2004

[34] Henry W. Ott; *"Noise Reduction Techniques in Electronic Systems"*, Second Edition, John Wiley & Sons, INC, pp. 29-42, 1988.

[35] *Hugh D. Young; Roger A. Freedman; A. Lewis Ford; "University Physics with Modern Physics"*, 12th Edition, pp. 918-925, Pearson Education Inc., Pearson Addison-Wesley, 2008.

[36] Henry W. Ott; *"Noise Reduction Techniques in Electronic Systems"*, Second Edition, John Wiley & Sons, INC, pp. 42-43, 1988.

[37] Henry W. Ott; *"Noise Reduction Techniques in Electronic Systems"*, Second Edition, John Wiley & Sons, INC, pp. 52-55, 1988.

[38] Henry W. Ott; *"Noise Reduction Techniques in Electronic Systems"*, Second Edition, John Wiley & Sons, INC, pp. 30-33, 1988.

[39] Henry W. Ott; *"Noise Reduction Techniques in Electronic Systems"*, Second Edition, John Wiley & Sons, INC, 1988.

[40] Peng-Yang Hung; Ying-Shu Lou; Yih-Lang Li; , "Minimum Shield Insertion on Full-Chip RLC Crosstalk Budgeting Routing," *Quality Electronic Design, 2008. ISQED 2008. 9th International Symposium on* , vol., no., pp.514-519, 17-19 March 2008.

[41] David Harris; *"Introduction to CMOS VLSI Design; Lecture 15: Non-ideal Transistors"*, Harvey Mudd College, p.22. Spring 2004.

[42] *Oregon State University; "CMOS Power Consumption Lecture 13"*, Electrical and Computer Engineering, ECE 322: Electronics 1, pp.1-35, Fall 2003.

[43] Masaharu Goto; James O. Barnes; Ronnie E. Owens; "*CMOS Programmable Delay Vernier",* Hewlett-Packard Journal, p. 56, October 1994.

[44] Kenneth S. Kundert; "*Power Supply Noise Reduction"*, Designer's Guide Consulting, INC. Version 4, p.3, January 2004.

[45] Kenneth S. Kundert; "*Power Supply Noise Reduction"*, Designer's Guide Consulting, INC. Version 4, p.2, January 2004.

[46] Farzan, S.M.D.; Ramahi, O.M.; , *"Effects of discrete bypass capacitors in Power/Ground planes with EBG structures,"* Electromagnetic Compatibility, 2008. EMC 2008. IEEE International Symposium on , vol., no., pp.1-6, 18-22 Aug.

[47] Dooghabadi, M.Z.; Hjortland, H.A.; Lande, T.S.; , *"High precision calibrated digital delay element," Electronics Letters* , vol.47, no.9, pp.564-565, April 28 2011.

[48] Figueiredo, M.J.; Aguiar, R.L.; , *"Noise and Jitter in CMOS Digitally Controlled Delay Lines,"* Electronics, Circuits and Systems, 2006. ICECS '06. 13th IEEE International Conference, vol., no., p.1357, 10-13 Dec. 2006

[49] Leroy Davis; *"Logic Family Noise Margin"*, Interfacebus, Jan. 17. 2012.

[50] Goran S. Jovanović; Mile K. Stojčev; "*Linear Current Starved Delay Element"*, Faculty of Electronic Engineering, Serbia and Montenegro, ICEST, pp. 1-4, 2005.

[51] Maymandi-Nejad, M.; Sachdev, M.; , *"A digitally programmable delay element: design and analysis,"* Very Large Scale Integration (VLSI) Systems, IEEE Transactions on , vol.11, no.5, p.872, Oct. 2003

[52] Masaharu Goto; James O. Barnes; Ronnie E. Owens; "*CMOS Programmable Delay Vernier",* Hewlett-Packard Journal, p. 54, October 1994.

[53] David A. Johns; Ken Martin; "*Analog Integrated Circuit Design"*, p.475, John Wiley & Sons, Inc., 1997.

[54] Masaharu Goto; James O. Barnes; Ronnie E. Owens; "*CMOS Programmable Delay Vernier",* Hewlett-Packard Journal, p. 53, October 1994.

[55] Ramakrishnan, V.; Balsara, P.T.; , "Very high precision Vernier delay line based CMOS pulse generator," Architecture, Circuits and Implementation of SOCs, 2005. DCAS '05. Proceedings of the 2005 IEEE Dallas/CAS Workshop: , vol., no., pp. 225- 228, 10 Oct. 2005.

[56] Julius O. Smith III; "*Physical Audio Signal Processing for Virtual Musical Instruments and Audio Effects"*, Center for Computer Research in Music and Acoustics (CCRMA), Department of Music, Stanford University, Chapter 3.5, Jul. 2, 2010.

[57] Maymandi-Nejad, M.; Sachdev, M.; , *"A digitally programmable delay element: design and analysis," Very Large Scale Integration (VLSI) Systems, IEEE Transactions on* , vol.11, no.5, p.877, Oct. 2003

[58] Kenneth S. Kundert; "*Power Supply Noise Reduction"*, Designer's Guide Consulting, INC. Version 4, p.8, January 2004.

[59] Maymandi-Nejad, M.; Sachdev, M.; , *"A digitally programmable delay element: design and analysis," Very Large Scale Integration (VLSI) Systems, IEEE Transactions on* , vol.11, no.5, p.876, Oct. 2003

[60] David A. Johns; Ken Martin; "*Analog Integrated Circuit Design"*, p.25, John Wiley & Sons, Inc., 1997.

[61] David A. Johns; Ken Martin; "*Analog Integrated Circuit Design"*, p.24, John Wiley & Sons, Inc., 1997.

[62] Christian Piguet; "*Low-Power CMOS Circuits, Technology; Logic Design and CAD tools"*, Taylor and Francis Group, LLC, Chapter 2.3.3: Bulk CMOS Design Solutions for Subthreshold Leakage, 2006.

[63] Lee Eng Han; Valerio B. Perez; Mark Lambert Cayanes; Mary Grace Salaber; "*CMOS Transistor Layout KungFu"*, EDA Utilities, pp.1-39, 2005.

[64] F. Maloberti; "*Layout of Analog CMOS Integrated Circuits Part 2: Transistors and basic Cells Layout",* Labratorio di Microsistemi Integrati, pp.1-38, Italy, 2012.

[65] David A. Johns; Ken Martin; "*Analog Integrated Circuit Design"*, p.119, John Wiley & Sons, Inc., 1997.

[66] University of Utah; "*Common Centroid Handout"*, ECE/CS 5720/6720: Analog Integrated Circuit Design, Department of Electrical and Computer Engineering, Spring 2010.

[67] Maxim IC; "*INL/DNL Measurements for High-Speed Analog-to-Digital Converters (ADCs)"*, Application note 283, Nov 20, 2001.